How to See Hidden Patterns in Metamaterials with Interpretable Machine Learning

Zhi Chen*[1], Alexander Ogren[2], Chiara Daraio[2], L. Catherine Brinson[1], and Cynthia Rudin[1]

[1] *Duke University*
[2] *California Institute of Technology*

Abstract

Metamaterials are composite materials with engineered geometrical micro- and meso-structures that can lead to uncommon physical properties, like negative Poisson's ratio or ultra-low shear resistance. *Periodic* metamaterials are composed of repeating unit-cells, and geometrical patterns within these unit-cells influence the propagation of elastic or acoustic waves and control dispersion. In this work, we develop a new interpretable, multi-resolution machine learning framework for finding patterns in the unit-cells of materials that reveal their dynamic properties. Specifically, we propose two new interpretable representations of metamaterials, called *shape-frequency features* and *unit-cell templates*. Machine learning models built using these feature classes can accurately predict dynamic material properties. These feature representations (particularly the unit-cell templates) have a useful property: they can operate on designs of higher resolutions. By learning key coarse scale patterns that can be reliably transferred to finer resolution design space via the shape-frequency features or unit-cell templates, we can almost freely design the fine resolution features of the unit-cell without changing coarse scale physics. Through this multi-resolution approach, we are able to design materials that possess target frequency ranges in which waves are allowed or disallowed to propagate (frequency bandgaps). Our approach yields major benefits: (1) unlike typical machine learning approaches to materials science, our models are interpretable, (2) our approaches leverage multi-resolution properties, and (3) our approach provides design flexibility.

1 Introduction

Metamaterials are structured, composite materials with physical properties that are derived from the properties of their constituent materials, and the geometry in which the constituent materials are arranged. One class of metamaterials, *periodic* metamaterials, are created as periodic repetitions of fundamental geometries, referred to as "unit-cells". By carefully arranging the constituent materials within the unit cells, periodic metamaterials can be engineered to have desirable properties for a variety of engineering applications. For example, compared to conventional composite materials, mechanical metamaterials can be surprisingly lightweight and strong [53, 65], can have controllable fracture toughness [22], and can be highly energy- and vibration-absorptive [3, 17, 35, 42, 43]. Metamaterials can also exhibit highly non-standard elastic behavior such as auxeticity (negative Poisson's ratio) [29], cloaking [8], significant anisotropy [5], and shear-normal coupling [20]. Similar unconventional properties have also been reported in optical and electro-magnetic metamaterials, like negative refractive index materials and optical cloaking [9]. The existence of uncommon mechanical properties of some metamaterials (such as negative effective mass

*Corresponding author. E-mail: zhi.chen1@duke.edu

Preprint, under review.

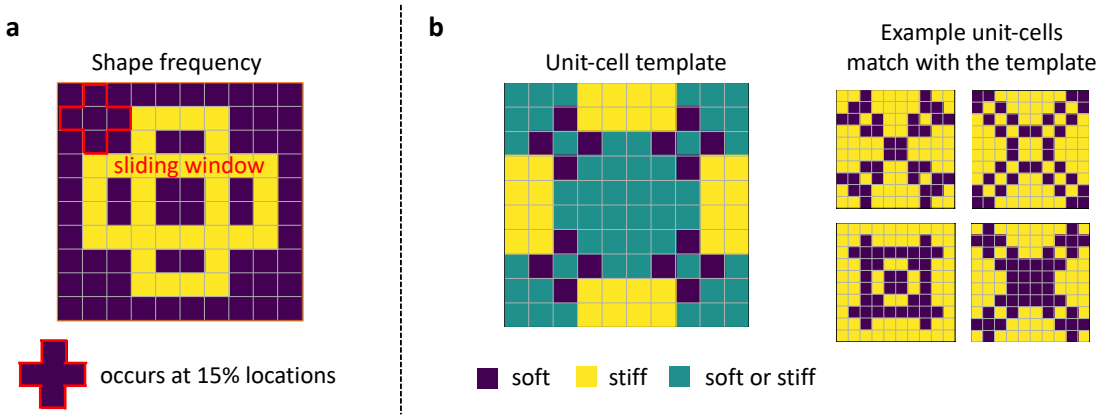


Figure 1: Examples of interpretable key patterns discovered by the proposed method. **a** A shape frequency feature (this one is shaped like a "+". How frequently this shape appears in the unit-cell is a useful predictor of a band gap. **b** A unit-cell template, which considers specific global patterns in the unit-cell. Here, regardless of whether we place stiff or soft materials at each green pixel in the unit-cell, as long as the stiff and soft materials are in the positions defined by the template in yellow and purple, there will be a band gap.

or stiffness [13, 15]) has been experimentally observed at different scales, which has prompted engineers to explore their use in different applications, ranging from materials for advanced batteries [62], to energy harvesting [12], seismic protection [7, 26, 46, 47], vibration or sound insulation [34, 41, 64], frequency filtering [48, 51], wave guiding [52, 57], heat management [40], RF telecommunication [10, 11], and ultrasonic imaging [28]. A fundamental aspect in the design of metamaterials is the ability to design the materials' basic building blocks (i.e., the unit-cells) to present specific properties.

The ability to control dynamic properties in metamaterials, such as propagation of sound signals and vibrations, is essential for many applications such as sound insulation (e.g., in buildings and concert halls), vibration insulation (e.g., airplanes and car components), and seismic wave redirection away from sensitive infrastructure. To control the propagation of elastic or acoustic waves, metamaterials rely on engineering the dispersion characteristics of the propagating waves. In specific examples, this can be done designing unit cells that *forbid specific frequency bands of propagation* (referred to as *band gaps*) in their dispersion relations, in which waves cannot propagate. Designing metamaterials with such properties, especially at higher physical and spacial dimensions can be challenging and computationally expensive.

Most metamaterials designed to date have been selected through a limited set of geometrical structures, often based on empirical trial-and-error or intuition [1]. Topology optimization has been used to design metamaterials [56, 55, 4, 1, 54], however the process is extraordinarily computationally expensive and is often limited to a single functionality. More recently, machine learning (ML) has been used for material design (and we will discuss this in depth later). A typical machine learning (ML) approach predicts material properties from unit cells defined by a finite set of pixels/voxels, and is used as a faster surrogate model for computationally expensive simulation. Such a black box approach does not provide insight into which geometric properties of materials are important. The lack of interpretability prevents scientists from gaining knowledge from the model or checking if the model is aligned with the specific materials' domain knowledge. In addition, typical machine learning would require a large simulation dataset, to predict target properties or perform inverse design accurately, which is computationally expensive. The simulation bottleneck

becomes even narrower as the design space becomes larger: the design space grows exponentially with the number of dimensions in the unit-cell, and thus the training data would also need to grow exponentially to avoid the curse of dimensionality in model fitting. We hypothesize this problem could be alleviated if the model can be trained on a small carefully-chosen subset of the design space, i.e., a coarser resolution design space, but generalize to the whole space, i.e., the finer resolution design space.

In this paper, we propose a novel interpretable multi-resolution ML approach for metamaterial design that has several advantages over the state-of-the-art black box ML approaches. In particular:

- ***Interpretability:*** The approach allows us to discover interpretable key patterns within unit-cells that are related to a physical property of interest (see Figure 1). We consider two types of patterns: (i) local patterns called *shape frequency features*, which calculate the occurrence frequency of certain shapes in the unit-cell; (ii) global patterns, called *unit-cell templates*, which look for arrangements of constituent materials in specific regions of the materials' unit-cells.

- ***Leverages Multi-resolution Properties:*** An important observation underpinning our methodology is that a pattern in the coarser resolution design space also exists in finer resolution design space, with one coarse pixel replaced by many finer pixels. As a result, *if a pattern can robustly characterize the target property at the coarse resolution design space, it will also be predictive at the finer resolution design space.* This leads to computationally-efficient discovery of many valuable metamaterial designs possessing the desired properties. In particular, our method allows us to construct a *scaffold of patterns* that allows interpretable coarse scale information discovered at low resolutions to be reliably transferred to make accurate predictions for high resolution designs.

- ***Flexible Metamaterial Designs:*** Our unit-cell templates (e.g., the one in Figure 1**b**) enables flexibility in unit-cell designs at any resolution. Simply, unit-cell templates specify regions where one can almost freely design unit-cell features without changing the target band gap property (in other words, by arranging constituent materials in the region of the template that our algorithm has marked in green). Such flexibility in design might be useful to satisfy practicality constraints such as connectivity.

Our multi-resolution approach is able to generate fine resolution metamaterials with target band gap properties without collecting massive amounts of new data at the finer resolution. Fine resolution simulations are computationally expensive, and typical ML approaches would simply start at the fine resolution and try to explore its (massive) space. However, by characterizing patterns at the coarse scale, our method suggests where we have freedom to design fine resolution features to achieve target wave behaviors.

Section 2 discusses the key problems of metamaterial design, and the limitations of current black box ML approaches in terms of solving these key problems. In Section 3, we introduce the four basic problem settings. In Section 4, we provide our two novel methods and explain how they deal with the four objectives. We then evaluate the proposed methods in Section 5, based on their performance with respect to the key problems mentioned in Section 2 and through tests on practical applications. We discuss and conclude in Section 6.

2 Key problems of metamaterial design

Unlike traditional ML settings, the goals of metamaterial design are not restricted to building an accurate prediction model. Here, we present four core objectives of data-driven approaches that are important to materials scientists: (a) design-to-structure prediction; (b) property-to-structure sampling; (c) identify key patterns; (d) transfer to finer resolution. More details about these four goals with visual illustration are shown in Figure 2.

Much past work on ML-based metamaterial design focuses on design-to-structure prediction (Objective (a)). Because of the expensive computational cost

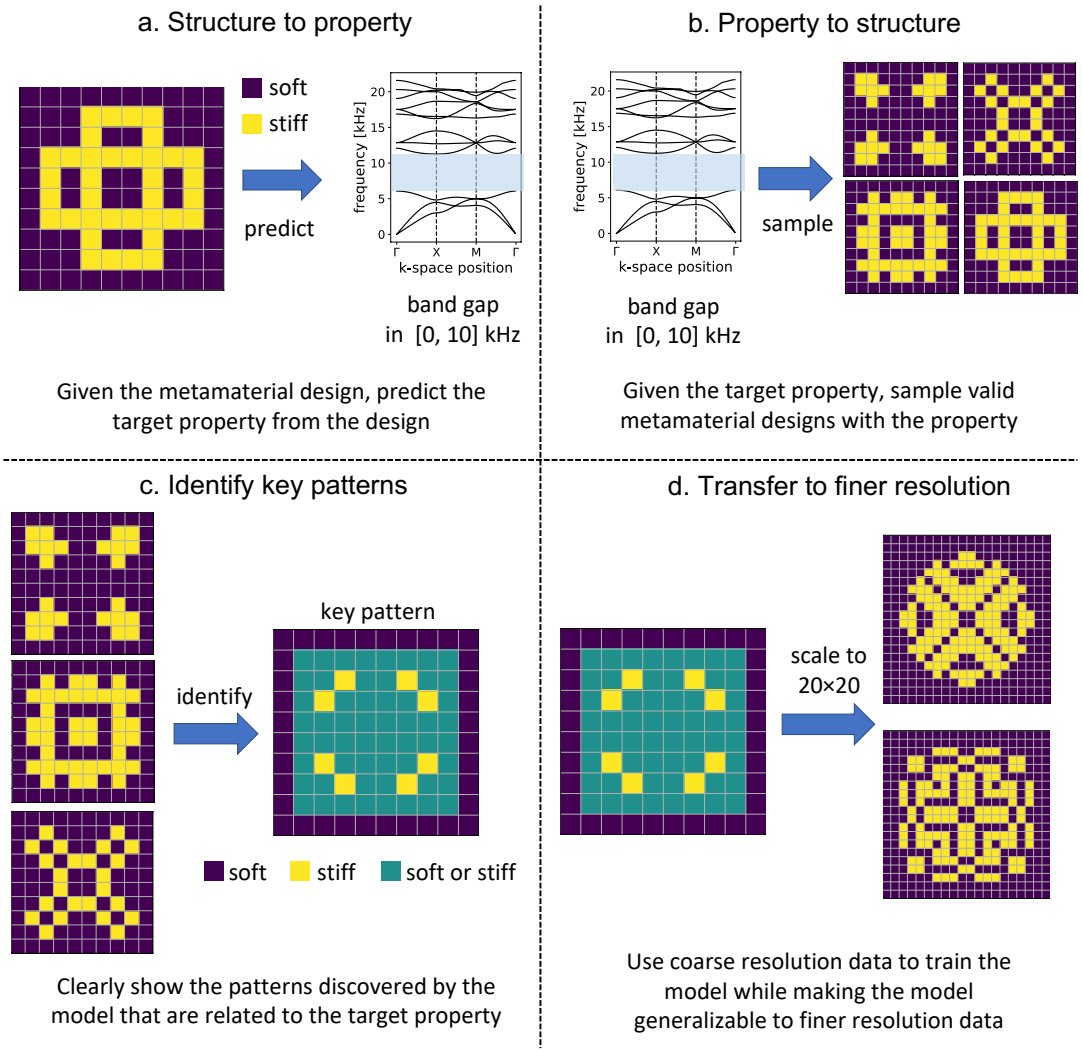


Figure 2: Core objectives of data-driven metamaterial designs.

of numerical simulations, these works train machine learning models (mostly neural networks) to approximate the simulation results, using these ML models as faster surrogates of the simulator [45, 6, 49, 18, 61]. Deep learning models require large labeled training data, but labels (i.e., material properties) are calculated by simulation, which is computationally expensive. Unlike our method, which alleviates this simulation bottleneck with a multi-resolution approach, [38] propose a self-supervised learning approach that can utilize randomly generated unlabeled data during training to use simulation results more efficiently. This method aims only at Objective (a), which is just the first step in the process of engineering a new metamaterial (and the approach is not interpretable).

The property-to-structure objective (Objective (b)) is an inverse problem with important practical outcomes: when materials with certain properties are needed, an algorithm ideally provides valid designs having these target properties. Such algorithms should leverage the results from Objective (a) to estimate the properties of each design, while the search is also constrained to consider valid designs.

Given the immense size of the design space, searching through the whole space is difficult or almost impossible. Advancements in deep generative models and neural network inverse modeling have allowed some recent works [33, 36, 16, 37, 58, 50, 32] to directly solve the inverse design problem for metamaterials, i.e., the property-to-structure objective. See [23, 39, 63] for reviews on deep learning methods addressing Objective (a) and (b) for metamaterial designs. There are two possible problems with these approaches. First, typical machine learning methods might suffer from curse of dimensionality, as we discussed. Second, from a scientific discovery perspective, solving the inverse design-to-structure problem using a black box is unsatisfying. Such approaches are unable to answer the key question, "What patterns in a material's design would lead to a specific desirable property?" A link between the specific design and the target property could be useful for further research; i.e., to determine whether there is an agreement with domain knowledge, and if not, to potentially discover new knowledge. The discovery of these interpretable patterns is Objective (c).

If we can transfer the knowledge gained from interpretable patterns discovered in Objective (c) from coarse to fine resolutions, it would give us a way to efficiently search the full space at the coarse resolution to determine what freedom we have in designing the fine resolution; this is Objective (d). In such a multi-resolution design process, a coarse resolution macro-structure can first be built for one property and then the finer resolution micro-structure can be fine-tuned to achieve another property simultaneously. The success of this multi-resolution approach depends only on robustly transferring of coarse scale information to finer resolution. As we will show the interpretable patterns, discovered by our method can be robustly transfered across resolutions. (Note that classically, deep learning methods require data to have a fixed input size, and we will show later, the traditional methods of resizing the finer resolution inputs work poorly in this case.)

The method proposed in this work simultaneously addresses all four objectives discussed above: By finding interpretable patterns (Objective (c)) that predict coarse scale physics well (Objective (a)) at fine resolutions (Objective (d)), we can engineer new materials with exotic properties (Objective (b)).

3 Problem Settings

Let us introduce the settings of the metamaterial design problem we are trying to solve, including the inputs and target of the dataset and their physical meanings.

We aim to design and characterize 2-D pixelated metamaterials made by tiling a 10×10 unit-cell. Such materials can be stacked to form 3D structures, and can direct, reflect or scatter waves, depending on the choice of unit-cell's material selection and geometry. In our framework, the unit-cell is a square with side length $a = 0.1$ m. However, transferring to a different length scale for a different application is easily doable by a simple scaling transformation on the dispersion relations. For a 10×10 unit-cell, the pixel side length is 1 cm; for 20×20 unit-cell, the pixel side length is 0.5 cm. Each unit-cell is made of two constituent materials: one is soft and lightweight, with elastic

modulus $E = 2$ GPa* and density $\rho = 1{,}000$ kg/m^3, and the other is stiff and heavy with $E = 200$ GPa, and $\rho = 8{,}000$ kg/m^3. These two sets of material properties are representative of a polymer and steel respectively. Our unit-cells are symmetric, with four axes of symmetry (x,y and $\pm 45°$). Under the symmetry constraints, the coarsest resolution (10×10) unit-cell has only 15 irreducible pixels. As a result, the raw input features of a sample in our dataset is a 15-dimensional binary vector: 0 means the soft constituent material in that location, and 1 means stiff constituent material. Thus, the full coarse space can be characterized, having 2^{15} total states. Figure 3 shows how to construct a material from the representation involving the 15 raw input features.

The material property we desire in our engineered materials is a band gap within a specific frequency range, given by the user. A band gap is a range of frequencies within which waves cannot propagate and are instead reflected.

To identify the existence of a band gap, one can examine the effect of dispersion in metamaterials by calculating dispersion relations. Wave dispersion occurs when waves of different wavelengths have different propagation velocities and dispersion arises in discrete systems, such as in metamaterials consisting of a periodic repetition of unit cells. Dispersion relations are functions that relate the wavenumber of a wave to its frequency, and they contain information regarding the frequency dependent propagation and attenuation of waves. Dispersion relations are found by computing elastic wave propagation solutions over a dense mesh of wavevectors. A band gap exists when there is a range of frequencies in the dispersion relation for which no wave propagation solutions exist (see Figure 2**a**,**b**).

Dispersion relation computations use Bloch-Floquet periodic boundary conditions, i.e., they assume that a given unit-cell is tiled infinitely in space. The physics revealed in dispersion analysis (infinite-tiling) can be leveraged in more realistic finite-tiling scenarios, as we will demonstrate later, which makes dispersion relation computations very useful for exploration and design of real materials. Our dispersion relation simulations are implemented using the finite element method.

More details about the simulation can be found in the Supplementary Information A.

We are looking for materials with band gaps in a certain frequency range. To define this task as a supervised classification problem, we create a binary label based on existence of a band gap in a given frequency range (e.g. [10, 20] kHz): 1 means one or more band gaps exist, 0 means no band gap exists. In other words, if a band gap range intersects with the target frequency range, the label is 1, otherwise the label is 0. One can also flexibly adjust the band gap label for different practical uses. For example, we can set the label to 1 only when the intersection of the band gap and the target frequency range is above a minimum threshold. We can also set the label to 1 when the band gap covers the entire target range, or even create a label for band gap properties in multiple frequency ranges.

4 Method

This section introduces methods proposed to realize the four objectives. Section 4.1 explains the shape-frequency features and how they can be used to optimize different objectives. Inspired by the efficiency of shape-frequency features, we propose unit-cell template sets, which make coarse-to-fine transitions easier, and thus is relevant for the fourth objective in Section 4.2.

4.1 Shape-Frequency Features

Denote the unit-cell as a $n \times n$ binary matrix $\mathbf{U} \in \{0,1\}^{n \times n}$, where $\mathbf{U}_{i,j} = 0$ means pixel i, j is assigned to the soft material and $\mathbf{U}_{i,j} = 1$ when the pixel is assigned to the stiff material. We propose an interpretable representation of unit-cells relying on *the fraction of the unit-cell in which specific small shapes are present*. These fractions are called *shape-frequency features*.

A specific shape s can be represented as a set of location offsets O_s whose elements are coordinates of pixels with respect to a reference pixel.

*GPa is gigapascals, a unit used to quantify elastic modulus.

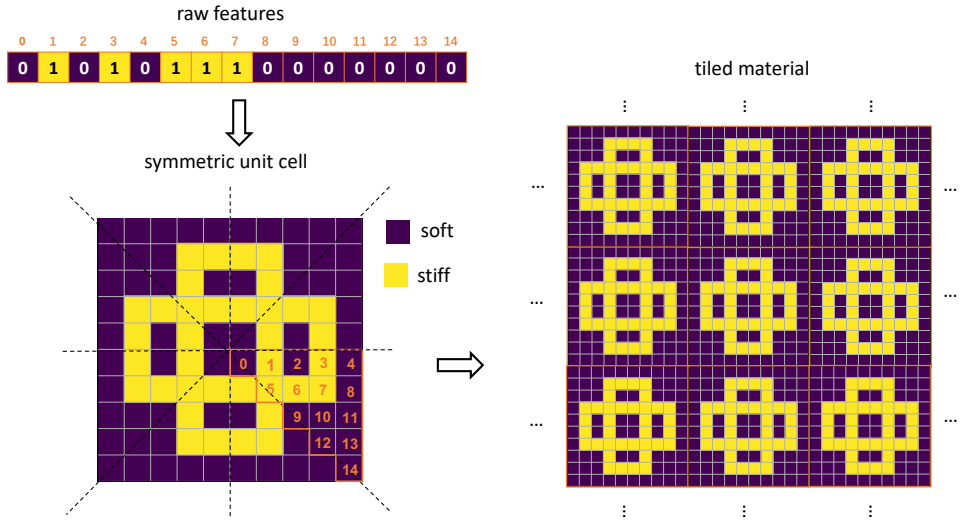


Figure 3: Constructing tiled 2-D material with the raw features. Upper Left: raw 15 dimensional feature vector. Lower left: the feature vector defines the triangle in the lower right of the unit-cell. The triangle is copied using lines of symmetry to define the full unit-cell. Right: the unit-cell is tiled to obtain the full material.For the Bloch-Floquet boundary conditions, the tiling is infinite in all dimensions.

For example, a 2×2 square window can be represented as $\{(0,0),(1,0),(0,1),(1,1)\}$. A 3×3 plus symbol as in Figure 1 can be represented as $\{(0,1),(1,0),(1,1),(1,2),(2,1)\}$. For a specific unit-cell, the feature value corresponding to that shape is computed by sliding the shape over the unit-cell and calculating the fraction of times that it is entirely contained within the soft material,

$$f_s = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbb{1} \left[\left(\sum_{(o_r,o_c) \in O_s} \mathbf{U}_{i+o_r,j+o_c} \right) = 0 \right], \tag{1}$$

where $\mathbb{1}[\cdot]$ is the indicator function. It equals 1 if and only if all the pixels in the shape are soft material (i.e., $\sum_{o_r,o_c} \mathbf{U}_{i+o_r,j+o_c} = 0$).

We would typically consider a collection of shapes, and have one element in a unit-cell's feature vector per shape. Thus, for unit-cell i, the jth component of its feature vector corresponds to how often the full shape j appears in its soft material.

In more detail, consider the collection of shapes, shown within Figure 4**a** (left). Note that this collection of shapes is the full set used in experiments, not just examples of them. Again, pixels of the stiff material are in yellow, and the soft material is shown in purple. We slide each of the shapes (sliding windows) over the unit-cell (Figure 4**a** (middle)) and count the fraction of positions over which the shape is fully contained within pixels of the soft (purple) material. These fractions together form the new representation for the unit-cell (Figure 4**a** (right)). Note that, to calculate the fraction, we should also consider the situation where the sliding window is across the boundary of two unit-cells, since the entire material is made by tiling the unit-cell. Because the unit-cells are symmetric, the occurrence of the patterns within the unit-cell are also symmetric: if we rotated the patterns by 90°, 180° or 270°, the number of detection of the pattern within the unit-cell would be identical. Note that the shape-frequency features are different from standard convolution filters used in computer vision; details are discussed in the Supplementary Information B. Theoretically, our method can be generalized to non-

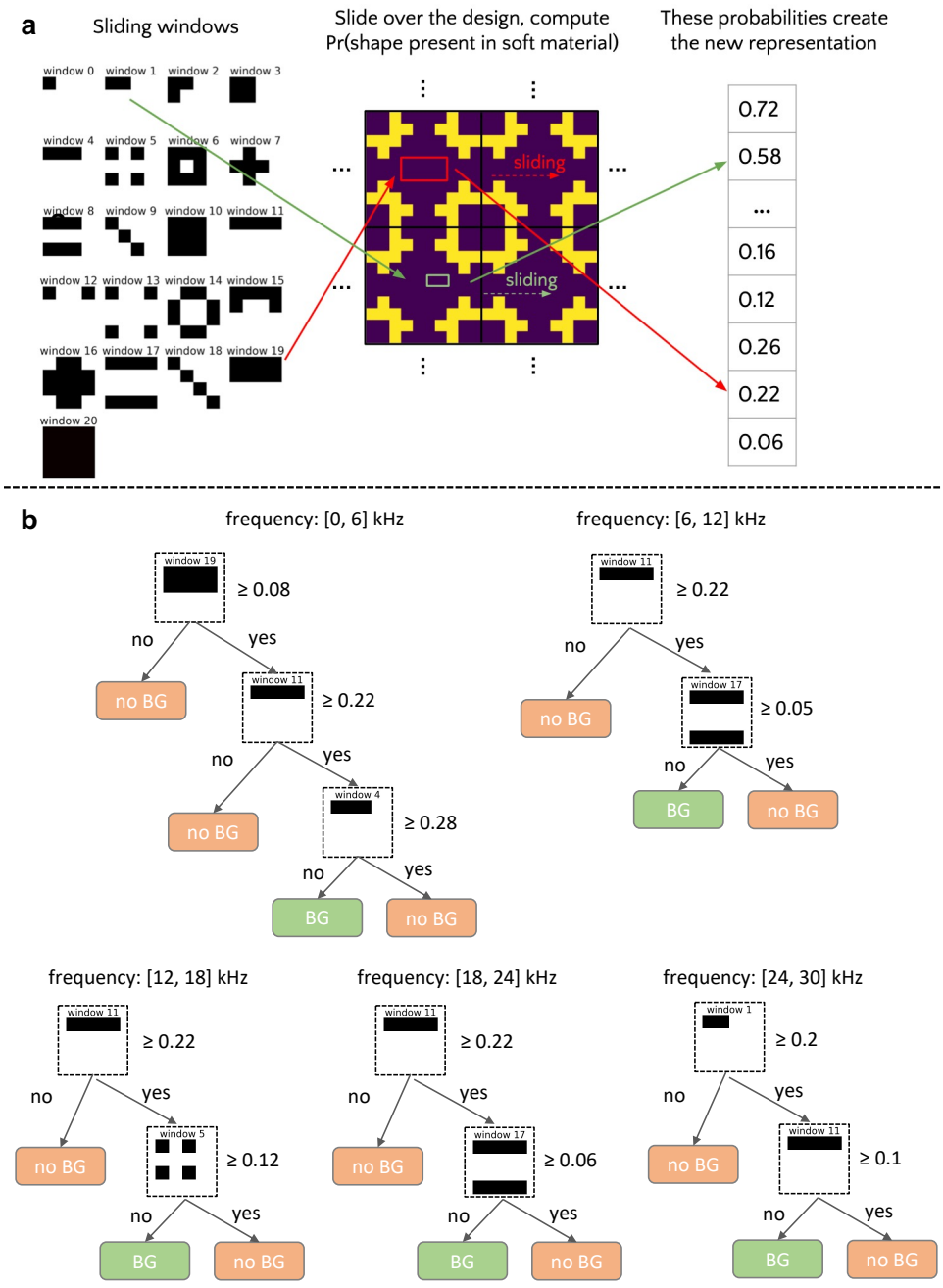


Figure 4: **a.** The process of calculating shape-frequency features. *Left*: Collection of shapes (sliding windows) used to create the shape-frequency features. *Middle:* An example 10×10 unit-cell design (tiled 4 times for better visualization). *Right:* Shape-frequency features of the unit-cell, which count the fraction of locations in the unit-cell where the shape is present in the soft material; **b.** Optimal sparse decision trees built on shape-frequency features for predicting band gaps in different frequency ranges.

square pixels and unit-cells as well, see Supplementary Information E.

We will show later that the shape-frequency features have a powerful ability to predict the existence of band gaps. With shape-frequency features, one can easily build an interpretable model (e.g., a sparse decision tree) that clearly shows how a pattern is related to the target property. Figure 4**b** shows examples of sparse decision trees built from shape frequency features.

4.1.1 Optimizing Precision and Support

For the property-to-structure task (the discovery of new designs with the target property), our objective for supervised learning will be a balance of *precision* (percentage of discovered designs with the target property) and *support* (total number of discovered designs), as opposed to standard metrics like accuracy or balanced accuracy. This is because we aim to find rules that capture a large number of unit-cell designs, and the unit-cells captured by those rules should each have a high chance of having the desired band gap. (We are not aiming for high recall, i.e., percentage of discovered designs among all designs with the target property, and we are not aiming to classify the full space, just to identify a promising part of it that we could explore.) Most machine learning methods cannot directly optimize custom objectives with constraints, such as precision, constrained by support. However, there are new approaches that permit direct optimization of custom discrete objectives. We use GOSDT (Generalized and Scalable Optimal Sparse Decision Trees, [31]) for this task, because it directly optimizes decision trees for customized objectives. We programmed it to maximize the following custom objective to optimality:

$$\max_{\text{tree}} \left[\frac{TP_{\text{tree}}}{TP_{\text{tree}} + FP_{\text{tree}} + \epsilon} - \frac{K}{TP_{\text{tree}} + \epsilon} \right] \tag{2}$$

$$= \max_{\text{tree}} \left[\frac{P - FN_{\text{tree}}}{P - FN_{\text{tree}} + FP_{\text{tree}} + \epsilon} - \frac{K}{P - FN_{\text{tree}} + \epsilon} \right]. \tag{3}$$

Here, K is a parameter that balances the precision and the support; ϵ is a small constant for numerical convenience; TP, FP, TN, FN means true positives, false postives, true negatives and false negatives. Equation (2) shows precision (first term) and inverse support (second term); we use inverse support so that if support is large, the term diminishes in importance. The simplification in Equation (3) shows that the objective is monotonically decreasing with respect to FN_{tree} and FP_{tree}. Theorem B.1 of [31] shows that as long as the objective is decreasing with respect to FN_{tree} and FP_{tree}, we can find an optimal sparse decision tree using GOSDT's branch and bound algorithm.

4.1.2 Property-to-structure Sampling

Because we optimize the precision using GOSDT, the false positive rate of our model will be low enough to work with. At this point, we can directly do rejection sampling for the property-to-structure problem without invoking the physics-based models: randomly sample a design and only accept the sample when the GOSDT model predicts the existence of a band gap. After sampling, each accepted sample is evaluated with the physics-based finite-element model to determine whether a band gap is present.

4.1.3 Transfer to Finer Resolution

The raw feature space of finer resolution samples is different from that of coarse resolution samples. Therefore, for the finer resolution data, we need to slightly modify the approach to obtain shape-frequency features that are compatible with the coarse resolution shape-frequency features. Suppose we want to transfer the model from 10×10 to 20×20 space. Then, there are three changes in calculating the shape-frequency features:

- The window size should be doubled when moving from coarse resolution to fine resolution;
- The stride of the sliding window should be 2 instead of 1;
- In counting the shape-frequency values, exact agreement between the window and soft material (purple) should be replaced with near exact agreement. In particular, when less than 2 yellow pixels (stiff material) are found in the window, we can consider this to be an agreement.

Algorithm 4.1 Sampling Fine resolution Unit-cell Designs via Shape-frequency Features

Input: simulated coarse resolution (10×10) dataset $\mathcal{D} := \{\mathbf{x}_i, y_i\}_{i=1}^{2^{15}}$,$\mathbf{x}_i \in \{0,1\}^{15}$: raw features; y_i: band gap label
Parameters: set of shapes S; tree sparsity regularization λ (see [31]); K, ϵ (see Section 4.1.1)
Output: raw features of a fine resolution unit-cell $\tilde{\mathbf{x}}$

calculate shape-frequency features $\mathbf{x}_i^{\text{SFF}} = \text{SFF}(\mathbf{x}_i, S)$ (coarse resolution), see Section 4.1
train an optimal sparse decision tree $\tau = \text{GOSDT}(\{\mathbf{x}_i^{\text{SFF}}, y_i\}_{i=1}^{2^{15}}, \lambda, K, \epsilon)$, see Section 4.1.2
while(True):
    randomly sample a binary vector $\tilde{\mathbf{x}}$ as raw features in the fine resolution space
    calculate shape-frequency features $\tilde{\mathbf{x}}^{\text{SFF}} = \text{SFF}(\tilde{\mathbf{x}}, S)$ (fine resolution), see Section 4.1.3
    if $\tau(\tilde{\mathbf{x}}^{\text{SFF}}) = 1$:
        return $\tilde{\mathbf{x}}$

When using the shape-frequency features with these modifications, the decision tree model learned on the coarse resolution data can be directly applied to the fine resolution. Thus, we can also do rejection sampling on the fine resolution with the model.

Algorithm 4.1 shows the entire pipeline of using an optimal sparse decision tree built on shape-frequency features to sample fine resolution designs with the target band gap property. Visual illustration of the sampling process can be found in Supplementary Information C.

4.2 Unit-cell Template Sets

Here, we introduce another interpretable machine learning model, called unit-cell template sets. Different from sparse trees on shape-frequency features, which focus on local patterns, a unit-cell template captures a global pattern for the unit-cell that is related to the target properties. The unit-cell template is a $n \times n$ matrix $\mathbf{T} \in \{0, 1, *\}^{n \times n}$, where $\mathbf{T}_{i,j} = 0$ means the pixel is soft material, $\mathbf{T}_{i,j} = 1$ means the pixel is stiff material, and $\mathbf{T}_{i,j} = *$ means the pixel could be either soft or stiff.

Definition *(match).* We say a unit-cell design $\mathbf{U}$ *matches* with the unit-cell template if and only if all pixels with value 0 on the template are also 0 on the design, and all pixels with value 1 on the unit-cell template are also 1 on the design. That is, $\forall(i,j)$ such that $\mathbf{T}_{i,j} \neq *$, we have $\mathbf{U}_{i,j} = \mathbf{T}_{i,j}$.

A unit-cell template set contains a set of unit-cell templates, and the sample design is predicted as positive iff it matches at least one unit-cell template in the set. Figure 5**a** shows an example of a unit-cell template set that consists of five different templates. The relationship between the unit-cell template set and other machine learning methods are discussed in the Supplementary Information B. Theoretically, our method can be generalized to nonsquare pixels and unit-cells as well, see Supplementary Information E.

The training objective of the model is to find a small number of unit-cell templates, such that the training precision of the entire model is high enough, and the model covers as many valid designs as possible, i.e., maximizing the support under a minimum precision constraint. The training process contains two steps, a pre-selection of candidate unit-cell templates (Section 4.2.1), and a integer linear programming (ILP) formulation to optimally select from the candidates (Section 4.2.2). Figure 5**b** shows the unit-cell template sets learned by the proposed algorithm for band gap prediction. More analysis of these discovered patterns can be found in Section 5.3.

4.2.1 Pre-selection of Templates

Here we consider symmetric unit-cell templates, because designs in our dataset are all symmetric. By removing all symmetry redundancy, the unit-cell template can be represented by its irreducible pixels, i.e.

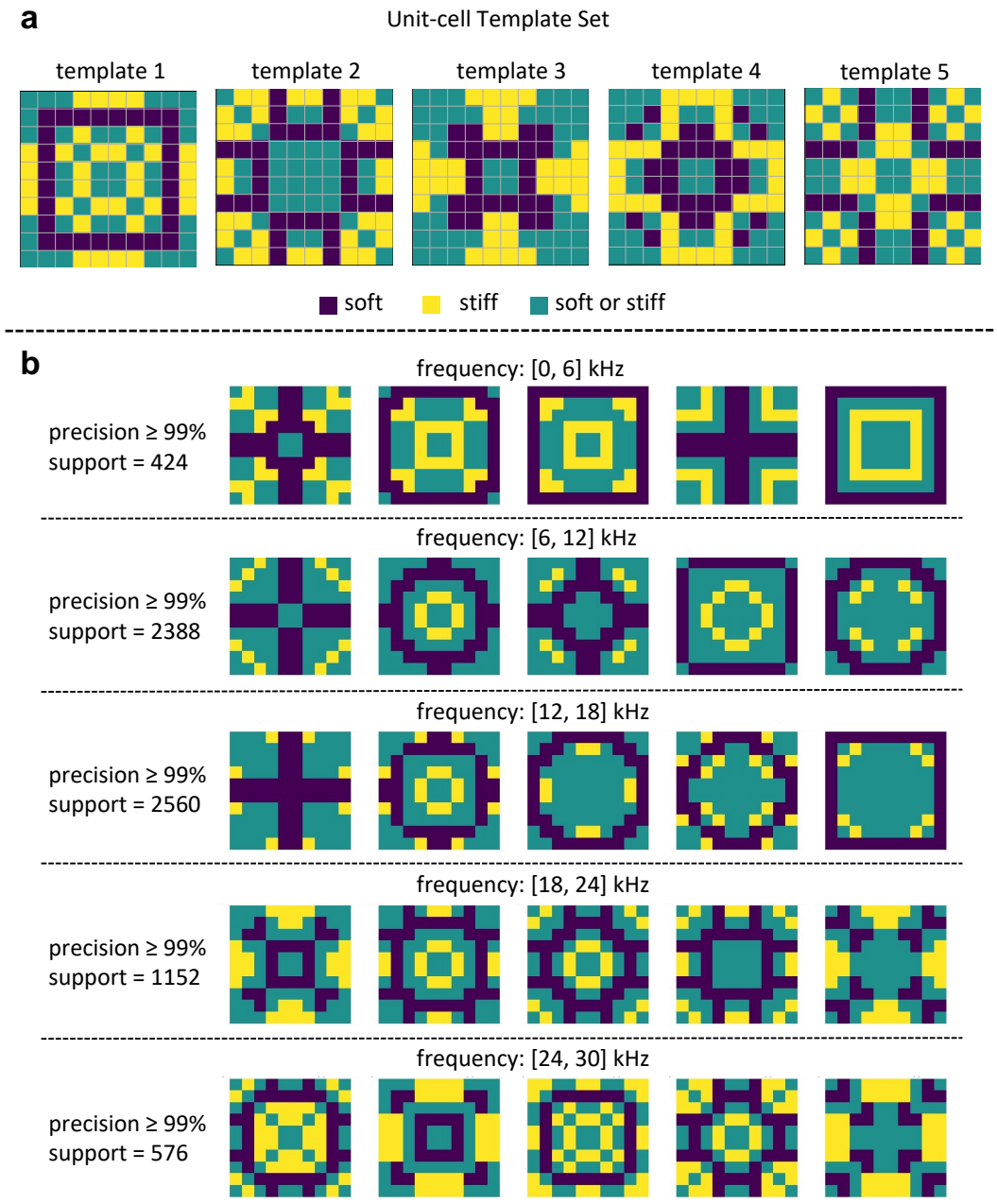


Figure 5: **a**. An Example of a unit-cell template set. A unit-cell is predicted as positive as long as it matches with at least one unit-cell template in the set; **b**. unit-cell template sets learned for predicting band gaps in different frequency ranges. Note that because unit-cells are tiled, a large cross through the center is identical to a square on the border. (E.g., consider the two upper right unit-cells.)

a 15 dimensional vector $\mathbf{t} \in \{0, 1, *\}^{15}$. The total number of possible templates is 3^{15} which is approximately 14.3 million. This is too large for the ILP in the next step. However, among the 3^{15} possible templates, most of them would never be selected because either their precision or support is not high enough. For example, a unit-cell template with precision 80% is not likely to be used if we want the entire model to have precision above 99%. Also, if the support of a unit-cell template, i.e., the total number of designs that match with it, is very small (e.g., < 10), the model may not generalize well. Therefore, we pre-select the unit-cell templates by setting minimum thresholds of precision and support, which reduces the search space only to promising unit-cell templates. We select the unit-cell template as a candidate only when it meets the minimum thresholds.

Observing that the entries are all binary in the unit-cell designs, we implement the precision calculation via bit operations, which significantly improves the speed of the pre-selection step. With the bit-operation implementation, the pre-selection steps of all 3^{15} possible templates can be finished in 50 seconds. The number of unit-cell templates that remains is typically in the range of 6000 to 12000, which is now suitable for ILP.

4.2.2 ILP for Template Selection

After the pre-selection step, we have a set of candidate unit-cell templates. Here, we formulate a ILP to optimally select from the candidates. Suppose we have n designs and m candidate templates. The goal of the ILP is to choose at most s unit-cell templates ($s \ll m$) whose union forms a model, such that the support is maximized and the precision of the model is at least p. Denote the true labels of all designs by a binary vector $\mathbf{y} \in \{0, 1\}^n$, and the predicted labels by $\hat{\mathbf{y}} \in \{0, 1\}^n$. $\mathbf{M} \in \{0, 1\}^{n \times m}$ denotes a matching matrix, where $M_{i,j}$ indicates whether design i matches with template j (1 for match, and 0 for no match). Binary vector $\mathbf{c} \in \{0, 1\}^m$ denotes the chosen unit-cell templates, where c_j indicates whether template j is chosen (1 for choose, and 0 for not choose). We solve the following ILP for template selection:

$$\max \sum_{i=1}^{n} \hat{y}_i \quad \text{(optimizing support)} \tag{4}$$

$$\text{s.t.} \sum_{j=1}^{m} c_j \leq s \quad \text{(sparsity constraint)} \tag{5}$$

$$\sum_{i=1}^{n} y_i \cdot \hat{y}_i \geq \left(\sum_{i=1}^{n} \hat{y}_i \right) \cdot p \quad \text{(minimum precision)} \tag{6}$$

$$\sum_{j=1}^{m} M_{i,j} \cdot c_j \geq \hat{y}_i, \quad i = 1, \ldots, n \quad \text{(define } \hat{y}_i\text{)} \tag{7}$$

$$\sum_{j=1}^{m} M_{i,j} \cdot c_j \leq m \cdot \hat{y}_i, \quad i = 1, \ldots, n \quad \text{(define } \hat{y}_i\text{)} \tag{8}$$

$$c_j \in \{0, 1\}, \quad j = 1, \ldots, m \tag{9}$$

$$\hat{y}_i \in \{0, 1\}, \quad i = 1, \ldots, n. \tag{10}$$

In this ILP, the objective (4) means maximizing the total number of designs predicted as positive, which is the same as support. Constraint (5) controls the sparsity, i.e., choose at most s unit-cell templates. Constraint (6) guarantees training precision of the model is at least p. (7) and (8) constraints together define $\hat{y}_i$, where $\hat{y}_i = 1$ iff design i matches with at least one of the chosen unit-cell templates. In particular, (7) says that if design i does not match any chosen template j (i.e., whenever c_j is 1, $M_{i,j}$ happens to be 0), then $\hat{y}_i$ will be set to 0. (8) will ensure that if there is a match for design i to any of the chosen templates (which all have $c_j = 1$), then this design is assigned $\hat{y}_i$=1. Using a commercial MIP solver, a problem with around 10000 candidate templates can be solved to optimality (when the current best solution meets the upper bound of the best possible solution) or near-optimality in about 10 minutes. If $s = 5$, it can be solved to optimality all the time, and when $s = 10$, the optimality gap (difference between current best solution and an upper bound of the best possible solution as the percentage of the upper bound) is always <20% for a run time of 30 minutes; it is worthwhile to note that optimal solutions are often attained quickly,

Algorithm 4.2 Sampling Fine Resolution Unit-cell Designs via Unit-cell Template Set

Input: simulated coarse resolution (10×10) dataset $\mathcal{D} := \{\mathbf{x}_i, y_i\}_{i=1}^{2^{15}}$, $\mathbf{x}_i \in \{0,1\}^{15}$: raw features; y_i: band gap label
Parameters: pre-selection support ψ_{pre}, pre-selection precision p_{pre}, sparsity constraint s, minimum precision p
Output: raw features of a fine resolution unit-cell $\tilde{\mathbf{x}}$
1: pre-select candidate template set S_T^{pre} = pre-selecting($\mathcal{D}$, $\{0,1,*\}^{15}$, ψ_{pre}, p_{pre}), see Section 4.2.1
2: run ILP to find optimal template set S_T^* = ILP($\mathcal{D}$, S_T^{pre}, s, p), see Section 4.2.2
3: randomly pick a template $\mathbf{t} \in S_T^*$
4: expand $\mathbf{t}$ to fine resolution space, get $\tilde{\mathbf{t}}$
5: randomly set to 0 or 1 for all * elements in $\tilde{\mathbf{t}}$, get $\tilde{\mathbf{x}}$
6: **return** $\tilde{\mathbf{x}}$

but the solvers can take a while to prove that the solution is optimal. Note that, if desired, one can set higher minimum precision and support thresholds for the pre-selection step to make the problem even smaller, so that the ILP can be solved even faster. We choose $s = 5$ for all the experiments in the main paper.

The result of the ILP is our unit-cell template set.

4.2.3 Property-to-structure Sampling

After training the design-to-structure model, the resulting unit-cell template set can be directly used to solve the inverse property-to-structure problem. An easy sampling procedure to do this is as follows: first, randomly choose a unit-cell template $\mathbf{t}$ from the unit-cell template set, where the probability to choose each template is proportional to its support; second, for all entries in $\mathbf{t}$ that equal *, randomly assign value 0 or 1 to them. These sampled unit-cells are likely to have the desired band gap.

4.2.4 Transfer to Finer Resolution

The unit-cell template set naturally transfers coarse scale information to finer resolutions. In particular, by subdividing each pixel in the unit-cell template into four sub-pixels, we directly obtain a unit-cell template defined on a finer-resolution space.

Algorithm 4.2 shows the entire pipeline of using unit-cell template set to sample fine resolution designs with the target band gap property. Visual illustration of the sampling process can be found in Supplementary Information C.

5 Experiments

The experiment section is organized according to the four objectives of data-driven metamaterial design discussed in the introduction. We evaluate how well the proposed methods can achieve these objectives, followed by several tests involving practical applications in materials discovery.

5.1 Objective 1: Design-to-structure Prediction

Let us test how well the proposed methods can perform design-to-structure prediction. We chose five frequency ranges ([0, 10], [10, 20], [20, 30], [30, 40] and [40, 50] kHz) to predict the existence of band gaps; these frequency ranges correspond to five different binary classification problems. Using balanced accuracy (bacc) as the evaluation metric, we compare the predictive performance of a diverse set of ML models with and without the shape-frequency features. We specifically consider linear models like support vector machines with linear kernels (SVMs) and logistic regression (LR); tree-based models like

Model		Frequency range				
		[0, 10]kHz	[10, 20]kHz	[20, 30]kHz	[30, 40]kHz	[40, 50]kHz
SVM	raw	71.77%	73.88%	50.85%	54.93%	49.84%
	SFF	75.62%	77.35%	67.96%	59.89%	49.93%
	improvement	+3.85%	+3.47%	+17.11%	+4.96%	0.09%
LR	raw	78.03%	75.44%	56.31%	76.59%	90.96%
	SFF	80.53%	79.55%	69.04%	76.9%	91.32%
	improvement	+2.50%	+4.11%	+12.73%	+0.31%	+0.36%
RF	raw	85.81%	80.04%	73.00%	77.66%	87.54%
	SFF	85.52%	81.98%	74.53%	81.26%	**95.35%**
	improvement	-0.29%	+1.94%	+1.53%	+3.60%	+7.81%
CART	raw	84.74%	75.68%	63.40%	73.95%	86.82%
	SFF	83.63%	80.13%	70.72%	79.73%	94.47%
	improvement	-1.11%	+4.45%	+7.32%	+5.78%	+7.65%
MLP	raw	90.72%	86.39%	78.15%	77.47%	65.90%
	SFF	85.90%	82.55%	76.01%	77.69%	66.4%
	improvement	-4.82%	-3.84%	-2.14%	+0.22%	+0.50%
LightGBM	raw	91.32%	88.27%	81.11%	82.62%	77.76%
	SFF	**96.10%**	**90.97%**	**85.57%**	**90.01%**	94.76%
	improvement	+4.78%	+2.70%	+4.46%	+7.39%	+17.00%
CNN	raw	93.24%	89.76%	81.65%	79.56%	84.83%

(a) Design-to-structure prediction: testing balanced accuracies (baccs) of different methods. We mark the models with the best bacc in each frequency range in bold.

Frequency range	SFF+GOSDT	Unit-Cell Template Sets	SFF+LightGBM	Raw+LightGBM
[0, 10]kHz	95.77%, 89	**98.53%**, 339	88.93%, 2169	80.77%, 2310
[10, 20]kHz	**98.11%**, 423	**98.68%**, 758	95.62%, 4063	93.52%, 4013
[20, 30]kHz	**94.15%**, 205	**94.08%**, 203	89.56%, 3811	86.94%, 3654

(b) Property-to-structure sampling: testing precision and support of different methods. Numbers in the table cells are formatted as "precision, support." The testing support here is calculated among 6554 testing samples (20% of the entire dataset).

Frequency range	CNN+resizing	LightGBM+resizing	SFF+GOSDT	Unit-Cell Template Sets		
	20×20	20×20	20×20	20×20	40×40	80×80
[0, 10]kHz	18.0%	25.0%	72.5%	**100.0%**	**100.0%**	**100.0%**
[10, 20]kHz	58.0%	68.5%	73.5%	**98.5%**	**99.0%**	**100.0%**
[20, 30]kHz	39.5%	52.5%	25.0%	**91.0%**	**96.0%**	**98.0%**

(c) Transfer to finer resolution: transfer precision of different methods.

Table 1: Summary of key quantitative results. The results are organized with respect to different objectives of data-driven metamaterials design. (a) Design-to-structure prediction; (b) Property-to-structure sampling; (c) Transfer to finer resolution.

CART, random forest (RF), and boosted trees (LightGBM [24]), as well as neural networks including the multi-layer perceptron (MLP). We also compare the proposed method with convolutional neural networks (CNNs), since they have been widely used in previous works of ML-based metamaterial design. The baccs of each model trained on raw feature, SFF, and improvements of SFF over raw features, are shown in Table 1 (a). We train each model 5 times and average the accuracy.

Our results show that **using the shape-frequency features, rather than the original raw features, improves the accuracy of classifiers for most machine learning methods**, especially tree-based methods such as boosted trees, but with the exception of MLP (SFF decreases its in [0, 10] kHz, [10, 20] kHz, and [20, 30] kHz).

One might expect CNNs to achieve great success in classifying band gaps for 2-D metamaterials since the unit-cells share many similarities with images. However, LightGBM [24] built on shape-frequency features outperforms ResNet18 [21] in all ranges. In some cases, e.g., within frequency range [40, 50] kHz, simple models like CART outperform CNNs.

More details of the experiment (e.g., hyperparameter settings) can be found in Supplementary Information D.1.

5.2 Objective 2: Property-to-structure Sampling

Using the methods discussed in Section 4.1.2 and Section 4.2.3, we are able to solve the inverse property-to-structure problem.

In practice, materials scientists need valid designs with the target property, but they do not require the set of *all* designs with the property. As such, our performance metric is precision, rather than recall. We also calculate the support, which is the total number of testing samples predicted as positive, to ensure the models can generate enough potentially-valid designs. Table 1 (b) lists the precision and support values from different methods. The methods we compared include GOSDT trained on shape-frequency features (denoted SFF) with the objective in Section 4.1.1, the unit-cell template set, and LightGBMs trained on SFF and raw features.

In terms of precision, **SFF+GOSDT and unit-cell template sets significantly outperformed LightGBMs**. This is probably owing to the fact that the proposed methods directly optimize precision. LightGBMs maintain larger support, while the support of SFF+GOSDT and unit-cell template sets is much lower. But for practical use, it is sufficient that the model finds dozens of valid designs. The average sampling time of these methods and the results of unit-cell template sets with different sparsity constraints can be found in Supplementary Information D.2.

5.3 Objective 3: Show Key Patterns

One advantage of the proposed methods is model interpretability; we aim to explicitly identify the key patterns learned from data that are related to the target property. In this way, domain experts can verify whether the learned rules are aligned with the domain knowledge, or even discover new knowledge.

In Figure 4**b** and Figure 5**b**, we visualize the GOSDT+SFF and unit-cell template set learned for band gaps in several frequency ranges ([0, 6], [6, 12], [12, 18], [18, 24], [24, 30] kHz).

In Figure 4**b**, the top splits of each tree trained on shape-frequency features seem to be looking for bars in the soft material. Conversely, the deepest nodes in the trees seem to identify the unit-cells that do not have as many soft material patterns. In the field of elastic wave propagation, it is known that the presence of stiff inclusions in a matrix of a softer material causes a Bragg-scattering effect, which may open a band gap. The trees we found seem to be looking for patterns that fit this description: the deepest node encourages the existence of stiff inclusion while the first node encourages more soft material.

The unit-cell templates (Figure 5**b**) contrast with the shape frequency features in that they explicitly identify global (rather than local) patterns. In the unit-cell template sets for different frequency ranges, we can observe the existence of soft circles (closed curves) and stiff inclusions inside the circles, which are responsible for the Bragg-scattering effect. As the

frequency range moves higher, the size of the circle decreases. This agrees with physical intuition that the smaller the stiff inclusions, the higher the frequency of the band gap.

5.4 Objective 4: Transfer to Finer Resolution

In Sections 4.1.3 and 4.2.4, we discussed how the proposed methods can transfer coarse scale information to finer resolution design space. To evaluate how well the model can transfer information, we trained the models on coarse resolution (10×10) unit-cells and tested them on finer resolution (20×20, 40×40 and 80×80) unit-cells. Table 1(c) shows the transfer precision of GOSDT+SFF and unit-cell template sets for band gaps in different frequency ranges. We also compared the proposed methods with two baselines: we resized the fine resolution (20×20) unit-cell to the original size (10×10) and applied two algorithms (CNN or LightGBM) for rejection sampling. As before, if the CNN or LightGBM model predicts that the resized design has a band gap, we accept that sample, otherwise we reject it. The resizing was done via bicubic interpolation. For each frequency range, we asked the trained models to sample 200 unit-cell designs in finer resolution space, and ran the FEA simulation to obtain the true band gap property for evaluating the transfer precision. For the baseline models and GOSDT+SFF, we show the results for 20×20 design space. As unit-cell template sets performs extremely well on this task, and generates new designs efficiently, we also show its results in 40×40 and 80×80 design space. Please see Supplementary Information C for visual illustrations of how to sample fine resolution designs using each model.

The results in Table 1(c) indicate that **the unit-cell templates, when transferred to all finer resolutions (**20×20**,** 40×40 **or** 80×80**), have very high precision, with almost no precision drop compared to** 10×10. GOSDT+SFF and other baselines do not generalize as well as unit-cell templates to the finer resolution design space. GOSDT+SFF performs better than the resizing baselines in [0, 10] kHz and [10, 20] kHz, but performs worse than baselines in [20, 30] kHz. Interestingly, the transfer precisions, of both GOSDT+SFF and unit-cell template sets, decrease as frequency ranges moves higher, although unit-cell template sets have a much slower precision decrease than GOSDT+SFF. A possible explanation for the decrease of precision is that, in higher frequency ranges, the band gaps are physically more related to finer scale features that are not included in the coarse resolution dataset. Since the models are trained on coarse resolution data, they can only transfer physics that occurs in coarse patterns to finer resolution design space, but cannot discover finer scale physics without supervision. But as shown by the transfer precision results, we should emphasize that our unit-cell template sets method was capable of extracting critical coarse resolution features such that this decrease of precision at finer resolution due to wave physics is minimized (the worst transfer precision is still above 90%). One further potential improvement to this is to add new samples at each finer resolution design space when transferring between extreme scales.

In Supplementary Information D.3, we show additional results on sampling with correlation between green pixels for unit-cell template sets, which demonstrates the surprising flexibility of unit-cell template sets in terms of designing at finer-resolution design space.

5.5 Practicality Test

In our method and simulations so far, we assumed the unit-cell is tiled infinitely for computational convenience. However, a unit-cell can only be tiled finitely in practice and the results can differ for infinite and finitely tiled domains due to boundary conditions. To test whether the designs found by our method work in practice, we simulated the dispersion relations of finitely-tiled materials made by unit-cell designs discovered by our method. See Supplementary Information D.4 for results of the finite tiling COMSOL simulation. The results show that our method is robust under finite tiling.

In addition to the finite tiling test, we also tested the practicality of the proposed method on its ability to create a wave demultiplexer (Figure 6), in which waves with different frequencies travel through the materials in different directions. That is, a signal

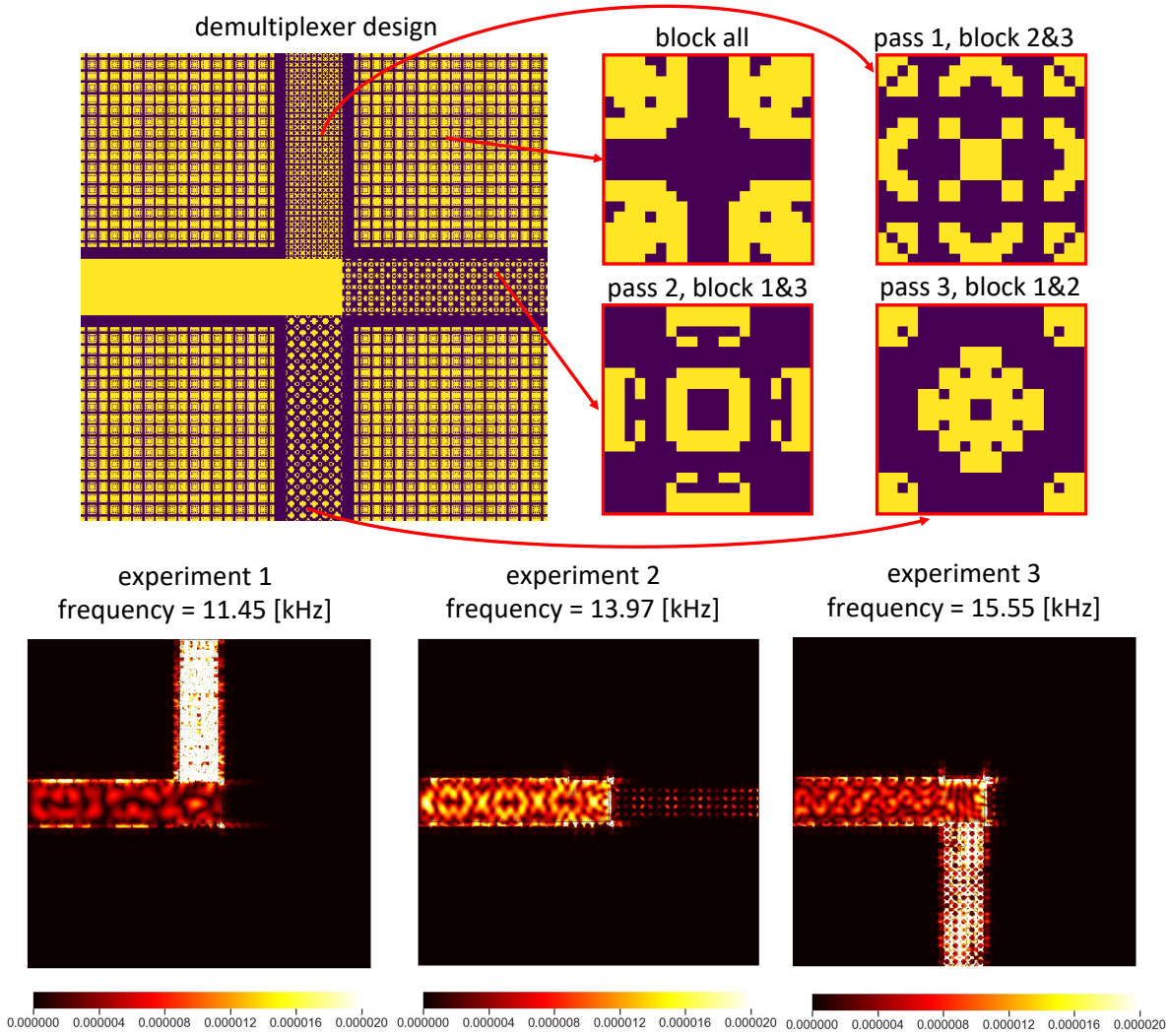


Figure 6: Mechanical wave demultiplexer and the displacement fields for input signals frequencies. Each part of the demultiplexer was built using our discovered unit-cells that have desired properties. *Top*: The demultiplexer and the unit-cell designs used to make the demultiplexer. *Bottom*: Magnitude of displacement fields when signals of different frequencies (11.45, 13.97 and 15.55 [kHz]) are fed into the left side of the demultiplexer. The displacement values are clipped if they exceed the display range. The signals with different frequencies go through different channels in the demultiplexer: 11.45 [kHz] goes upward, 13.97 [kHz] goes right, and 15.55 [kHz] goes downward, as desired. Note that, although the passing signal of experiment 2 is not as strong as signals in the other experiments, it still pass the channel on the right without fading inside the channel.

enters the demultiplexer, and there are three different possible outputs; which one will be non-zero depends on the frequency of the input signal. Specifically, we will build the demultiplexer to route signals from three different frequency ranges in different directions.

We need 5 different materials to build the demultiplexer: one material with band gaps covering all three ranges, one material allowing band pass in all ranges, and three materials allowing band pass in one range while blocking the other two ranges. We use homogeneous stiff unit-cells for the material allowing band pass in all ranges. For other materials, we train a unit-cell template set to find 20×20 unit-cell designs with these properties, and assemble the 5 unit-cells to build the demultiplexer (top row of Figure 6). The bottom row of Figure 6 shows the how the demultiplexer successfully guides waves with different frequencies (11.45, 13.97 and 15.55 [kHz]) towards different directions, which was the goal of the experiment.

6 Conclusion and Discussion

Our work shows the power of interpretable machine learning tools in material design. The approach has achieved both mechanistic understanding, e.g., physically interpretable rules of patterns that lead to band gaps, and designing new materials with desired functionality. The approach has been demonstrated to be predictive for both infinite domains and realistic finite domains, and it has been able to design the material geometry for a wave demultiplexer. Additionally, our multi-resolution framework, which robustly carries coarse-scale knowledge to finer resolutions, is potentially applicable to a wide range of materials science problems.

Since our method learns robust coarse-scale features that can generalize to finer-resolution design space, it might also be useful in future studies for determining how to collect finer-resolution training data for rapidly capturing fine-scale physics. Using these new data, we might be able to fine tune the model so that it can more efficiently capture physics at multiple scales.

Data Availability

All datasets and related simulation code will be available upon acceptance.

Code Availability

The code for replicating our experiments will be available upon acceptance.

References

[1] K. Bertoldi, V. Vitelli, J. Christensen, and M. van Hecke. Flexible mechanical metamaterials. *Nature Reviews Materials*, 2:17066, 2017.

[2] Jacob Bien and Robert Tibshirani. Prototype Selection for Interpretable Classification. *The Annals of Applied Statistics*, pages 2403–2424, 2011.

[3] O. R. Bilal, D. Ballagi, and C. Daraio. Architected lattices for simultaneous broadband attenuation of airborne sound and mechanical vibrations in all directions. *Physical Review Applied*, 10:054060, 2018.

[4] O. R. Bilal and M. I. Hussein. Ultrawide phononic band gap for combined in-plane and out-of-plane waves. *Physical Review E*, 84:065701, 2011.

[5] O. R. Bilal, R. Susstrunk, C. Daraio, and S. D. Huber. Intrinsically polar elastic metamaterials. *Advanced Materials*, 29, 2017.

[6] Ramin Bostanabad, Yu-Chin Chan, Liwei Wang, Ping Zhu, and Wei Chen. Globally approximate gaussian processes for big data with application to data-driven metamaterials design. *Journal of Mechanical Design*, 141(11), 2019.

[7] Stéphane Brûlé, EH Javelaud, Stefan Enoch, and Sébastien Guenneau. Experiments on seismic metamaterials: molding surface waves. *Physical Review Letters*, 112(13):133901, 2014.

[8] T. Bückmann, M. Thiel, M. Kadic, R. Schittny, and M. Wegener. An elasto-mechanical unfeelability cloak made of pentamode metamaterials. *Nature Communications*, 5:4130, 2014.

[9] W. Cai and V. Shalaev. *Optical Metamaterials: Fundamentals and Applications*. Springer-Verlag, New York, 2010.

[10] J. Cha and C. Daraio. Electrical tuning of elastic wave propagation in nanomechanical lattices at mhz frequencies. *Nature Nanotechnology*, 13:1016–1020, 2018.

[11] J. Cha, K. W. Kim, and C. Daraio. Experimental realization of on-chip topological nanoelectromechanical metamaterials. *Nature*, 564:229–233, 2018.

[12] Z. Chen, B. Guo, Y. Yang, and C. Cheng. Metamaterials-based enhanced energy harvesting: A review. *Physica B: Condensed Matter*, 438:1–8, 2014.

[13] J. Christensen, M. Kadic, O. Kraft, and M. Wegener. Vibrant times for mechanical metamaterials. *MRS Communications*, 5:453–462, 2015.

[14] Peter Clark and Robin Boswell. Rule induction with CN2: Some recent improvements. In *European Working Session on Learning*, pages 151–163. Springer, 1991.

[15] S. A. Cummer, J. Christensen, and A. Alù. Controlling sound with acoustic metamaterials. *Nature Reviews Materials*, 1:16001, 2016.

[16] Yang Deng, Simiao Ren, Kebin Fan, Jordan M Malof, and Willie J Padilla. Neural-adjoint method for the inverse design of all-dielectric metasurfaces. *Optics Express*, 29(5):7526–7534, 2021.

[17] P. Deymier. *Acoustic Metamaterials and Phononic Crystals*, volume 173. Springer-Verlag, Berlin Heidelberg, 2013.

[18] David Finol, Yan Lu, Vijay Mahadevan, and Ankit Srivastava. Deep convolutional neural networks for eigenvalue problems in mechanics. *International Journal for Numerical Methods in Engineering*, 118(5):258–275, 2019.

[19] Eibe Frank and Ian H Witten. Generating accurate rule sets without global optimization. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 144–151, 1998.

[20] T. Frenzel, M. Kadic, and M. Wegener. Three-dimensional mechanical metamaterials with a twist. *Science*, 358:1072, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[22] S. S. Injeti, C. Daraio, and K. Bhattacharya. Metamaterials with engineered failure load and stiffness. *Proceedings of the National Academy of Sciences*, 116:23960, 2019.

[23] Jiaqi Jiang, Mingkun Chen, and Jonathan A Fan. Deep neural networks for the evaluation and design of photonic devices. *Nature Reviews Materials*, pages 1–22, 2020.

[24] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.

[25] Been Kim, Cynthia Rudin, and Julie A Shah. The Bayesian Case Model: A Generative Approach for Case-Based Reasoning and Prototype Classification. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, volume 27, pages 1952–1960, 2014.

[26] S.-H. Kim and M. P. Das. Seismic waveguide of metamaterials. *Modern Physics Letters B*, 26(12501):05, 2012.

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.

[28] S. Krödel and C. Daraio. Microlattice metamaterials for tailoring ultrasonic transmission with elastoacoustic hybridization. *Physical Review Applied*, 6, 2016.

[29] R. Lakes. Foam structures with a negative poisson's ratio. *Science*, 235:1038, 1987.

[30] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684, 2016.

[31] Jimmy Lin, Chudi Zhong, Diane Hu, Cynthia Rudin, and Margo Selzer. Generalized optimal sparse decision trees. In *Proc. International Conference on Machine Learning*, 2020.

[32] Zhaocheng Liu, Dayu Zhu, Kyu-Tae Lee, Andrew S Kim, Lakshmi Raju, and Wenshan Cai. Compounding meta-atoms into metamolecules with hybrid artificial intelligence techniques. *Advanced Materials*, 32(6):1904790, 2020.

[33] Zhaocheng Liu, Dayu Zhu, Sean P Rodrigues, Kyu-Tae Lee, and Wenshan Cai. Generative model for the inverse design of metasurfaces. *Nano Letters*, 18(10):6570–6576, 2018.

[34] F. Ma, J. H. Wu, M. Huang, W. Zhang, and S. Zhang. A purely flexible lightweight membrane-type acoustic metamaterial. *Journal of Physics D: Applied Physics*, 48(17510):5, 2015.

[35] G. Ma and P. Sheng. Acoustic metamaterials: From local resonances to broad horizons. *Science Advances*, 2, 2016.

[36] Wei Ma, Feng Cheng, and Yongmin Liu. Deep-learning-enabled on-demand design of chiral metamaterials. *ACS nano*, 12(6):6326–6334, 2018.

[37] Wei Ma, Feng Cheng, Yihao Xu, Qinlong Wen, and Yongmin Liu. Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy. *Advanced Materials*, 31(35):1901111, 2019.

[38] Wei Ma and Yongmin Liu. A data-efficient self-supervised deep learning model for design and characterization of nanophotonic structures. *SCIENCE CHINA Physics, Mechanics & Astronomy*, 63(8):1–8, 2020.

[39] Wei Ma, Zhaocheng Liu, Zhaxylyk A Kudyshev, Alexandra Boltasseva, Wenshan Cai, and Yongmin Liu. Deep learning for the design of photonic structures. *Nature Photonics*, 15(2):77–90, 2021.

[40] M. Maldovan. Sound and heat revolutions in phononics. *Nature*, 503:209–217, 2013.

[41] J. Mei, G. Ma, M. Yang, Z. Yang, W. Wen, and P. Sheng. Dark acoustic metamaterials as super absorbers for low-frequency sound. *Nature Communications*, 3:756, 2012.

[42] M. Mohsenizadeh, F. Gasbarri, M. Munther, A. Beheshti, and K. Davami. Additively-manufactured lightweight metamaterials for energy absorption. *Materials & Design*, 139:521–530, 2018.

[43] J. Mueller, K. H. Matlack, K. Shea, and C. Daraio. Energy absorption properties of periodic and stochastic 3d lattice materials. *Advanced Theory and Simulations*, 2(19000):81, 2019.

[44] COMSOL Multiphysics. Introduction to COMSOL multiphysics®. *COMSOL Multiphysics, Burlington, MA, accessed Feb*, 9:2018, 1998.

[45] Christian C Nadell, Bohao Huang, Jordan M Malof, and Willie J Padilla. Deep learning for accelerated all-dielectric metasurface design. *Optics express*, 27(20):27523–27535, 2019.

[46] A. Palermo, S. Krödel, A. Marzani, and C. Daraio. Engineered metabarrier as shield from seismic surface waves. *Scientific Reports*, 6:39356, 2016.

[47] A. Palermo, S. Krödel, K. H. Matlack, R. Zaccherini, V. K. Dertimanis, E. N. Chatzi, A. Marzani, and C. Daraio. Hybridization of guided surface acoustic modes in unconsolidated granular media by a resonant metasurface. *Physical Review Applied*, 9:054026, 2018.

[48] Y. Pennec, B. Djafari-Rouhani, J. O. Vasseur, A. Khelif, and P. A. Deymier. Tunable filtering and demultiplexing in phononic crystals with hollow cylinders. *Phys Rev E Stat Nonlin Soft Matter Phys*, 69:046608, 2004.

[49] Tianshuo Qiu, Xin Shi, Jiafu Wang, Yongfeng Li, Shaobo Qu, Qiang Cheng, Tiejun Cui, and Sai Sui. Deep learning: a rapid and efficient route to automatic metasurface design. *Advanced Science*, 6(12):1900128, 2019.

[50] Haoran Ren, Wei Shao, Yi Li, Flora Salim, and Min Gu. Three-dimensional vectorial holography based on machine learning inverse design. *Science advances*, 6(16):eaaz4261, 2020.

[51] C. J. Rupp, M. L. Dunn, and K. Maute. Switchable phononic wave filtering, guiding, harvesting, and actuating in polarization-patterned piezoelectric solids. *Applied Physics Letters*, 96(11190):2, 2010.

[52] C. J. Rupp, A. Evgrafov, K. Maute, and M. L. Dunn. Design of phononic materials/structures for surface wave devices using topology optimization. *Structural and Multidisciplinary Optimization*, 34:111–121, 2007.

[53] T. A. Schaedler, A. J. Jacobsen, A. Torrents, A. E. Sorensen, J. Lian, J. R. Greer, L. Valdevit, and W. B. Carter. Ultralight metallic microlattices. *Science*, 334:962–965, 2011.

[54] C. Schumacher, B. Bickel, J. Rys, S. Marschner, C. Daraio, and M. Gross. Microstructures to control elasticity in 3d printing. *ACM Transactions on Graphics (TOG)*, 34:136, 2015.

[55] O. Sigmund. In systematic design of metamaterials by topology optimization. In Iutam Symposium, editor, *on Modelling Nanomaterials and Nanosystems, Dordrecht, 2009; Pyrz, R*, pages 151–159. Dordrecht, Rauhe, J. C., Eds. Springer Netherlands, 2009.

[56] Ole Sigmund and Jakob Søndergaard Jensen. Systematic design of phononic band–gap materials and structures by topology optimization. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 361(1806):1001–1019, 2003.

[57] M. Torres, Montero de Espinosa, F. R., D. García-Pablos, and N. García. Sonic band gaps in finite elastic media: Surface states and localization phenomena in linear and point defects. *Physical Review Letters*, 82:3054–3057, 1999.

[58] Liwei Wang, Yu-Chin Chan, Faez Ahmed, Zhao Liu, Ping Zhu, and Wei Chen. Deep generative modeling for mechanistic-based learning and design of metamaterial systems. *Computer Methods in Applied Mechanics and Engineering*, 372:113377, 2020.

[59] Tong Wang, Cynthia Rudin, Finale Doshi, Yimin Liu, Erica Klampfl, and Perry MacNeille. Bayesian Or's of And's for interpretable classification with application to context aware recommender systems. In *International Conference on Data Mining (ICDM)*, 2016.

[60] Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. A bayesian framework for learning rule sets for interpretable classification. *Journal of Machine Learning Research*, 18(70):1–37, 2017.

[61] Peter R Wiecha and Otto L Muskens. Deep learning meets nanophotonics: a generalized accurate predictor for near fields and far fields of arbitrary 3d nanostructures. *Nano letters*, 20(1):329–338, 2019.

[62] J. Xiao, J. Zheng, X. Li, Y. Shao, and J. G. Zhang. Hierarchically structured materials for lithium batteries. *Nanotechnology*, 24(42400):4, 2013.

[63] Yihao Xu, Xianzhe Zhang, Yun Fu, and Yongmin Liu. Interfacing photonics with artificial intelligence: an innovative design strategy for photonic structures and devices based on artificial neural networks. *Photonics Research*, 9(4):B135–B152, 2021.

[64] Z. Yang, H. M. Dai, N. H. Chan, G. C. Ma, and P. Sheng. Acoustic metamaterial panels for sound attenuation in the 50–1000 Hz regime. *Applied Physics Letters*, 96:041906, 2010.

[65] X. Zheng, H. Lee, T. H. Weisgraber, M. Shusteff, J. DeOtte, E. B. Duoss, J. D. Kuntz, M. M. Biener, Q. Ge, J. A. Jackson, S. O. Kucheyev, N. X. Fang, and C. M. Spadaccini. Ultralight, ultrastiff mechanical metamaterials. *Science*, 344:1373–1377, 2014.

[66] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals (Seventh Edition)*. Oxford, Butterworth-Heinemann, 2013.

Acknowledgements

The authors are grateful to M. Bastawrous, A. Lin, K. Liu, C. Zhong, O. Bilal, W. Chen, C. Tomasi, and S. Mukherjee for the feedback and assistance they provided during the development and preparation of this research. The authors acknowledge funding from the National Science Foundation under grant OAC-1835782, Department of Energy under grant DE-SC0021358, and National Research Traineeship Program under NSF grant DGE-2022040.

Competing Interests

The authors declare no competing interests.

A Simulation Settings

In-house/Infinite-medium Simulator

To determine whether a unit-cell design has a bandgap, we evaluate the dispersion relation along the wavevector contour (Irreducible Brillouin Zone contour) shown in Figure S1. To evaluate the dispersion relation, we solve a series of eigenvalue problems given by the harmonic elastic wave equation with Bloch-Floquet periodic boundary conditions.

Dispersion relation analysis is performed only on the domain of a single unit-cell, and assumes that the metamaterial has a unit cell that is infinitely repeating in each direction that Bloch-Floquet periodic boundary conditions are applied.

The harmonic elastic wave equation is:

$$\rho(x)\omega^2 u = \frac{\partial}{\partial x_j}\left[C_{ijkl}(x)\frac{1}{2}\left(\frac{\partial u_k}{\partial x_l}+\frac{\partial u_l}{\partial x_k}\right)\right] \quad i,j,k,l = 1,2. \tag{11}$$

where C is the material stiffness tensor, ρ is the material density, u is the displacement field (eigenfunction), ω is the eigenfrequency, and x represents positions in the 2-D space. Displacements (and therefore wave modes) are confined to the same 2-D plane that contains the metamaterial. Bloch-Floquet theory says that the displacement field solutions, u, should be spatially periodic, so we introduce the wavevector γ to characterize this spatial periodicity. To enforce Bloch-Floquet periodicity, we apply Bloch-Floquet periodic boundary conditions:

$$u(x+a_n) = u(x)e^{-\mathbf{i}\gamma\cdot a_n} \;\forall x \in \{x \in \Omega \mid x + a_n \in \Omega\}, \quad n = 1,2. \tag{12}$$

where $\mathbf{i}$ is the imaginary unit, a_n are the lattice vectors representing the periodicity of the material, and $\Omega \subset \mathbb{R}^2$ represents the domain of the unit-cell.

To discretize and solve this eigenvalue problem, we apply the Finite Element method using bilinear quadrilateral elements. The resulting discretized eigenvalue problem is

$$[K(\gamma) - \omega^2 M(\gamma)]u = 0, \tag{13}$$

where K and M are the stiffness and mass matrices, respectively. The boundary conditions are baked into the wavevector-dependent stiffness and mass matrices. This code is implemented in *MATLAB*.

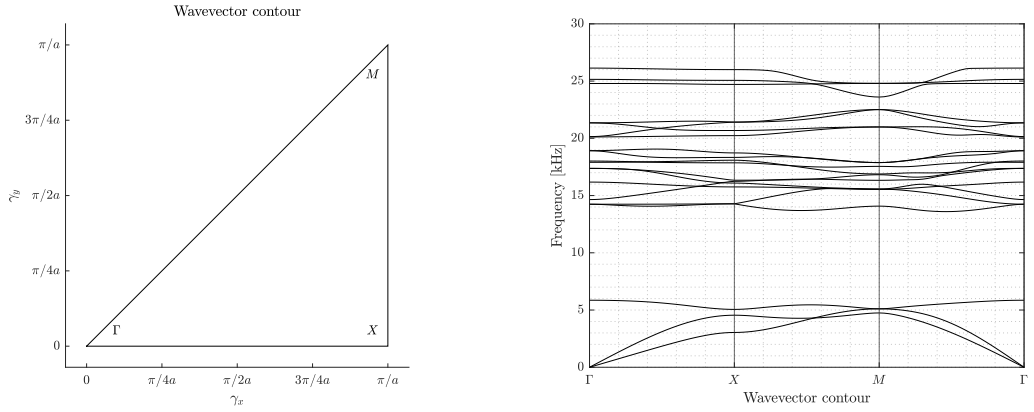


Figure S1: Left: Wavevector contour (also known as Irreducible Brillouin Zone contour). Right: Example of a dispersion relation evaluated along the wavevector contour.

Commercial/Finite-medium Simulator

Dispersion relation analysis is performed only on the domain of a single unit-cell, and assumes that the metamaterial has a unit cell that is infinitely repeating in each direction that Bloch-Floquet periodic boundary conditions are applied. However, real periodic metamaterials do not extend infinitely in any directions. While dispersion analysis is a critically useful method for converging on a unit-cell design with desirable properties, it is important to check that these properties are still present when the metamaterial is only finitely periodic.

To perform this validation, we perform frequency domain analysis on finitely periodic geometries using the commercial Finite Element software *COMSOL Multiphysics* [66] with the Structural Mechanics Module.

Material Properties

The material is a 2-D metamaterial made by tiling a 10×10 pixelated unit-cell. Each unit-cell is made of two constituent materials: one is soft and light, with elastic modulus and density of $E = 2[GPa]$, $\rho = 1,000[kg/m^3]$ respectively, and the other is stiff and heavy with $E = 200[GPa]$, and $\rho = 8,000[kg/m^3]$.

B Relationship of Proposed Methods to Other Machine Learning Methods

Comparison of shape-frequency features to CNN filters: One might think that the sliding windows of the shape-frequency features are similar to convolution filters in CNNs, but they are not. Shape-frequency features count only exact matches between each window and the soft constituent material, while a traditional convolutional layer would output a real number for the degree of match. In order to attempt to replicate the calculation of shape-frequency with a CNN, one would concatenate a convolutional layer, a Rectified Linear Unit (ReLU) layer and a linear layer in a CNN [27]: the convolutional layer calculates the matching score, the ReLU layer sets a threshold to round the matching score to 0 or 1, and the linear layer averages the rounded score over all locations. However, if we train such a CNN, the weights cannot be integers and thereby the model cannot be as sparse and interpretable as our method using the shape-frequency features.

Comparison of unit-cell template sets to prototype learning and decision sets: The unit-cell template set technique is closely related to prototype-based machine learning models (e.g., [2, 25]). The prototype-based models learn a set of prototypical cases (which can be whole observations or parts of observations) from the training data. Given a test observation, the model makes a decision by comparing the test observation to the prototypical cases. For example, in classification, the model would output the label of the the prototypical case that most closely resembles the test observation. A unit-cell template can be viewed as a part-based prototype because it is part of a training observation, and the decision of the model is made by matching the templates and the test observation. A difference between classical prototype-based learning and our approach is that classical objectives are classification losses while our method optimizes support and precision.

A unit-cell template set can also be viewed as an instance of a decision set. A decision set (e.g., [14, 19, 59, 30, 60]), is a logical model comprised of an unordered collection of rules, where each rule is a conjunction of conditions. In other words, the entire model is a disjunctive normal form (DNF), or "OR of ANDs." A positive prediction is made if at least one of the rules is satisfied. One unit-cell template in the set is essentially a conjunction of conditions that compares whether the pixel in the template is the same as the pixel in the test observation. That is, there is an "OR" over unit-cell templates and an "AND" over pixels in the template. As far as we know, decision sets have not previously been utilized for material discovery.

C Visual Illustration of Sampling in Finer Resolution Design Space

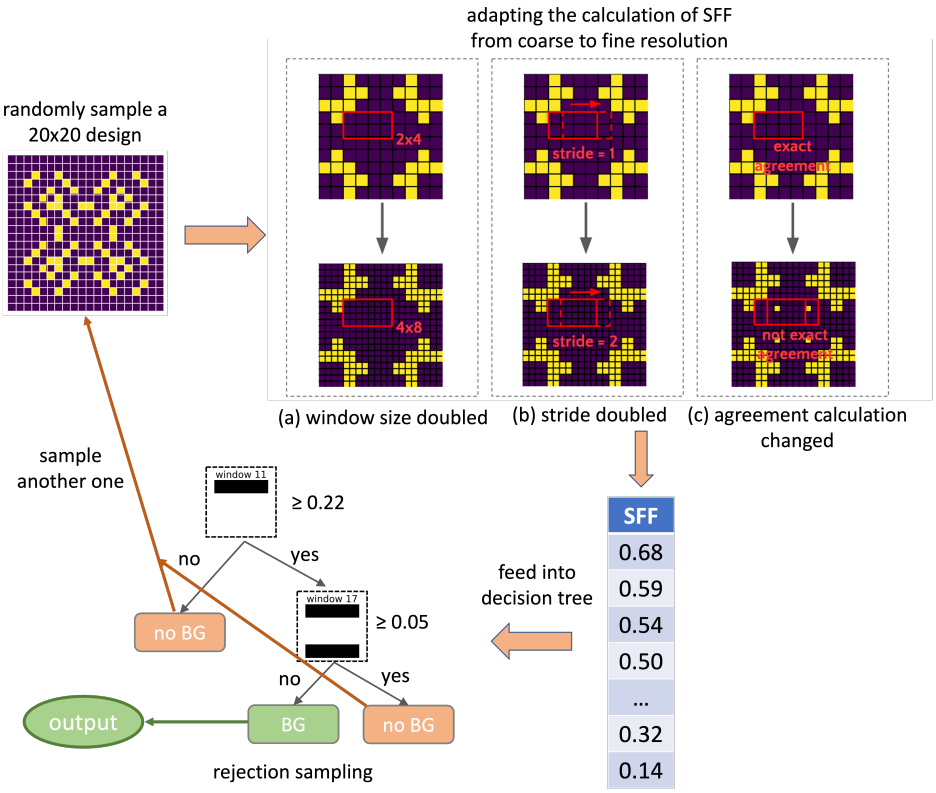
Figure S2(a) shows an example of how to sample a 20×20 design with GOSDT+SFF trained on 10×10 data. This is essentially a rejection sampling process, where a random 20×20 design is selected, and accepted if the algorithm predicts existence of a band gap, and rejected if not. However, as the design space has changed from 10×10 to 20×20, the calculation of SFF has also changed, with details discussed in Section 4.1.3 in the main text.

Figure S2(b) shows an example of how to sample a 20×20 design with a 10×10 unit-cell template. While still following the templates learned from coarse resolution (10×10) data, most of these finer resolution designs can never be found in the coarse resolution (10×10) design space.

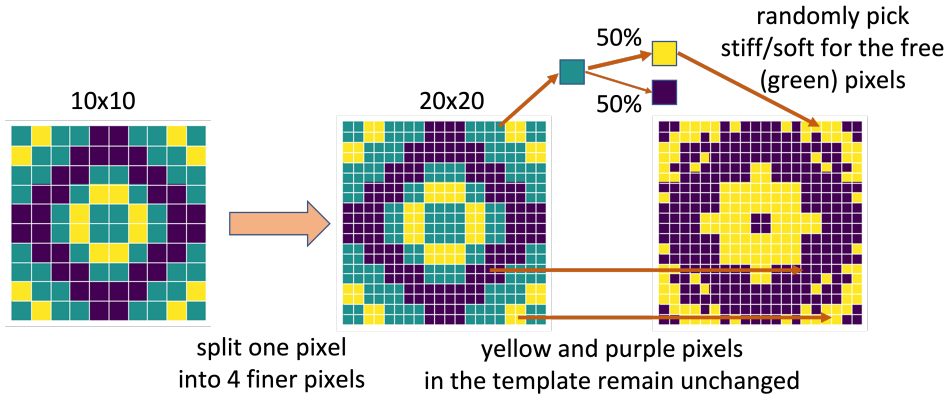
D More Experimental Results

D.1 Structure-to-property - Prediction Accuracy

We now compare the testing balanced accuracy (bacc) of various machine learning methods, including linear models like support vector machines with linear kernels (SVMs) and logistic regression (LR); tree-based models like CART, random forest (RF), and boosted trees (LightGBM [24]); as well as neural networks including the multi-layer perceptron (MLP) and convolutional neural networks (CNNs). LightGBM [24] was used with 300 trees, and the MLP had 2 hidden layers, each containing 100 neurons. The CNN architecture is ResNet18 [21] adapted to one-channel input, and default parameters were used for other methods. All other models were trained on shape-frequency features (SFF) and raw features respectively, while the CNN was trained on the entire 2-D design. We trained each model 5 times to average out randomness in the training process. Figure S3 shows the balanced accuracy results. Models trained on shape-frequency features generally perform much better than models trained on raw features, with the exception of MLP. Also, LightGBM trained on SFF outperforms CNN.

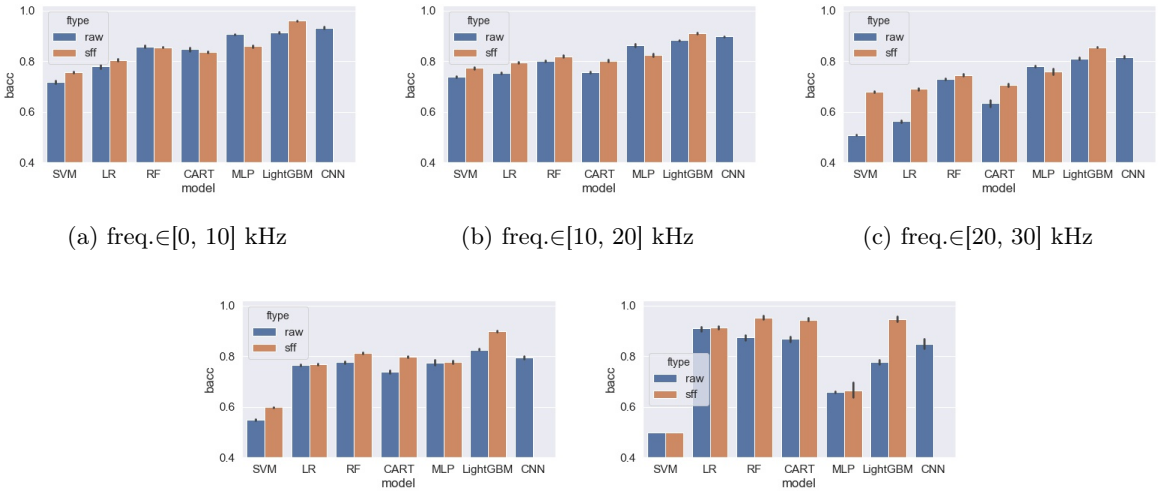


(a) Sampling finer resolution design with SFF+GOSDT trained on coarse resolution data. Sampling consists of rejection sampling with GOSDT, and adapting the calculation of SFF from coarse to fine resolution (see Section 4.1.3).



(b) Sampling finer resolution design with unit-cell template trained on coarse resolution data.

Figure S2: Visual illustration of how to sampling finer resolution design with proposed methods.



(a) freq.∈[0, 10] kHz (b) freq.∈[10, 20] kHz (c) freq.∈[20, 30] kHz

(d) freq.∈[30, 40] kHz (e) freq.∈[40, 50] kHz

Figure S3: Testing balanced accuracies (bacc) of different machine learning models. The label for each machine learning problem is in its subfigure caption, for instance, freq.∈[10, 20] kHz means predicting whether the material has a band gap within any frequency in the range [10, 20] kHz. Using the shape-frequency features (sff) generally improve performance over raw features. Note that it is not natural to design a convolution over the shape-frequency features since there is no notion of adjacency, so there is no CNN value for the shape-frequency features.

D.2 Property-to-structure Sampling - Precision and Support

SFF+GOSDT	Unit-Cell Template Sets	SFF+LightGBM	Raw+LightGBM
1.8×10^{-4} s	1.9×10^{-7} s	3.0×10^{-4} s	5.1×10^{-6} s

Table S1: Average sampling time of a new 10×10 design.

Table S1 compares the average time per sample drawn for each method. **The unit-cell template set has the fastest sampling speed** because the model explicitly represents all designs that match with it, while other methods rely on rejection sampling to solve the inverse problem. Models trained on shape-frequency features have a slower sampling rate than models trained on raw features, since calculation of shape-frequency features takes extra time. Also, due to their sparsity, GOSDT models have faster sampling than LightGBM. However, all these the sampling times work fine in practice.

Table S2 compares the performance of unit-cell template sets with different sparsity constraints ($s = 5$, $s = 8$, and $s = 10$). The MIP corresponding to each model was run with a time limit of 60 minutes. The best solution at that expiration time was used if the MIP was not solved to optimality. The results suggest that there is no significant precision difference when using different numbers of templates in the model. The support, i.e., the number of designs covered by the template set, increases as the number of templates increase. In practice, a support of $s = 5$ tends to be good enough.

Frequency range	$s = 5$	$s = 8$	$s = 10$
[0, 10] kHz	98.53%, 339	98.54%, 481	97.73%, 529
[10, 20] kHz	98.68%, 758	98.60%, 1142	98.35%, 1274
[20, 30] kHz	94.08%, 203	94.77%, 287	95.33%, 343

Table S2: Testing precision and support of unit-cell template sets with different sparsity constraints. s is the number of templates. Numbers in the table cells are formatted as "precision, support." The testing support here is calculated among 6554 testing samples (20% of the entire dataset).

Frequency range	no covariance	Matern 3/2 ($l = 2$)	Matern 3/2 ($l = 6$)	Matern 3/2 ($l = 10$)
[0, 10] kHz	100.0%	98.0%	96.5%	96.5%
[10, 20] kHz	98.5%	98.5%	96.5%	98.0%
[20, 30] kHz	91.0%	92.5%	95%	94%

Table S3: Transfer precision from coarse resolution (10×10) to finer resolutions with different sampling methods (20×20) in the free regions of the unit cell templates.

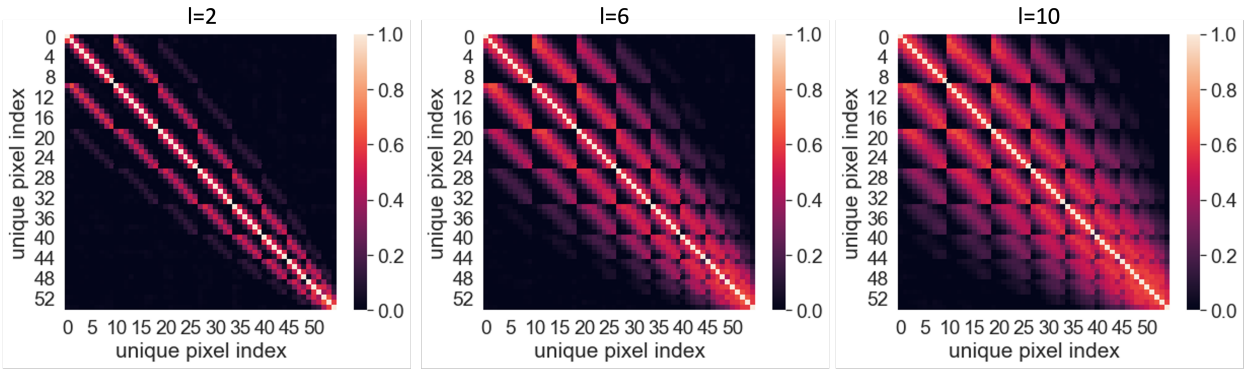
D.3 Transfer to Finer Resolution - Transfer Precision

In order to test the flexibility of unit-cell template sets in sampling finer scale designs, in addition to sampling the pixels independently in the free region of a unit-cell template, as we did in the main paper, we also try to sample pixels with correlation between them. Specifically, we use a Matern $\frac{3}{2}$ kernel, i.e.,

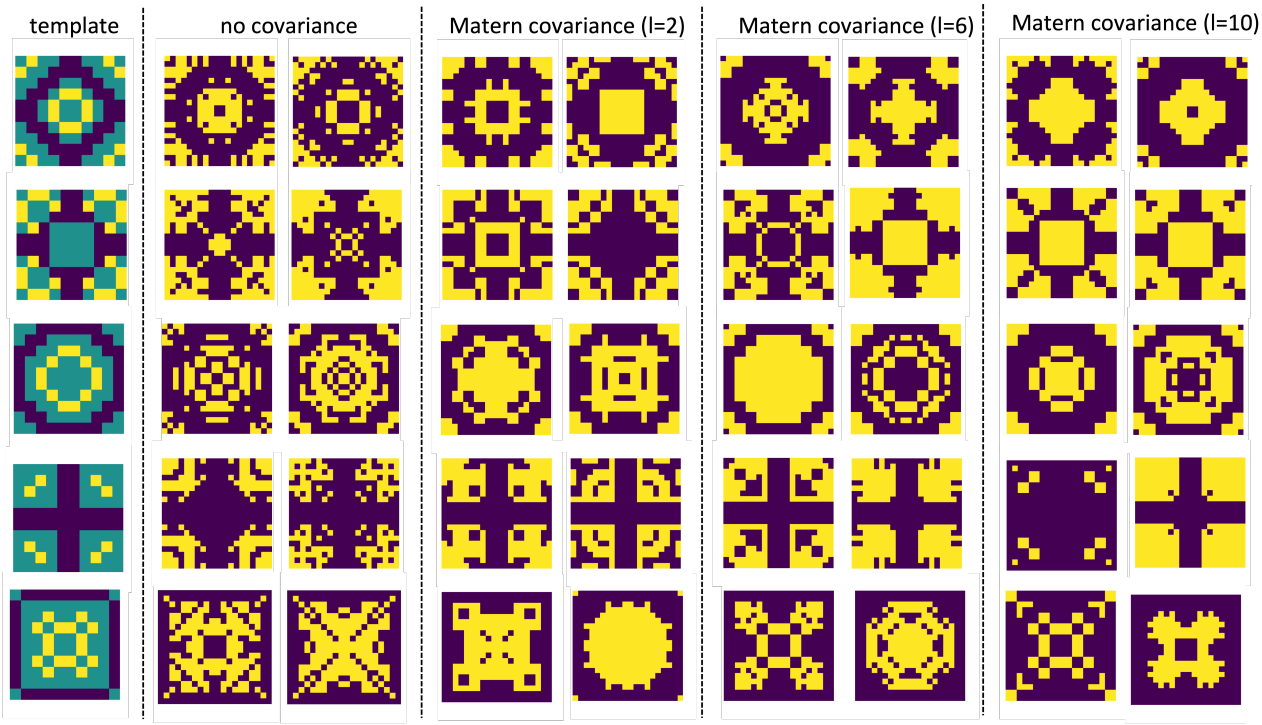
$$k(\mathbf{x}, \mathbf{x}') = (1 + \sqrt{3} \cdot \frac{d(\mathbf{x}, \mathbf{x}')}{l}) \cdot e^{-\sqrt{3} \cdot \frac{d(\mathbf{x}, \mathbf{x}')}{l}} \tag{14}$$

to calculate the covariance between two pixels $\mathbf{x}$ and $\mathbf{x}'$; d is the distance function, which is set to be the distance between the two pixels in the 2D design space; l is the scale parameter, the larger l is, the more the correlation spreads throughout the larger design region of the unit cell. Note that since the design pixels are binary, we cannot directly sample them with a given covariance matrix. Instead, we first rescale the covariance matrix elements by a transformation function $g(x) = \sin(\frac{x\pi}{2})$, then sample from a multivariate Gaussian distribution, and take the sign of the Gaussian variables to get the binary variables/pixels. Figure S4(a) shows the covariance matrices with different scale parameters, and Figure S4(b) shows the designs whose free pixels (green region of templates) are sampled with (i) no covariance, and (ii) the Matern 3/2 covariance matrices with $l = 2$, $l = 6$, and $l = 10$. Generally, the designs sampled with Matern covariance matrices are more regularized than independent sampling — independent designs look like coarse templates with fine white noise, while the Matern covariance designs are more similar to designs generated by humans. Also, larger values of the length scale parameter l would create larger length scale features (regions with larger areas of purple or yellow).

Table S3 shows the precision of these 20×20 designs. Interesting phenomena about wave physics can be founding when comparing the precisions in different frequency ranges and feature sizes of the generated designs. In the lowest frequency range, i.e., [0, 10] kHz, the transfer precision is the lowest at $l = 10$, when the generated designs have largest feature length scales. Here, length scale of feature means the average size of regions with the same color. In the highest frequency range, designs with the smallest feature length scale (independent fine pixels) produce the lowest precision. In the middle frequency range [10, 20] kHz, precision is the lowest at $l = 6$, which corresponds to medium feature length scales. This agrees with intuition about



(a) Matern 3/2 covariance matrices with different scale parameter ($l = 2$, $l = 6$, and $l = 10$). The larger l is, the more correlated the design pixels are.



(b) 20×20 designs sampled from unit-cell template set for band gaps in [0, 1000] using different sampling methods. From left to right are the template, pixels sampled with no covariance (independently), Matern 3/2 covariance ($l = 6$), Matern 3/2 covariance ($l = 2$), and Matern 3/2 covariance ($l = 10$)

Figure S4: Sampling with correlation. (a) covariance matrices; (b) sampled finer resolution (20×20) designs

band gap wave physics — band gaps in higher frequency ranges are more closely associated with finer-scale features, and thus would naturally more highly affected by finer-scale features generated in the green regions of unit-cell templates. Surprisingly, the unit cell templates still perform well (lowest precision $> 90\%$) even considering the interplay between wavelength and length scale of feature. These results indicate that our unit cell templates are very robust and flexible for material design. Given a template, materials scientists can design features almost freely in the green region to satisfy practical needs (e.g., connectivity constraints) while still maintaining the desired band gap.

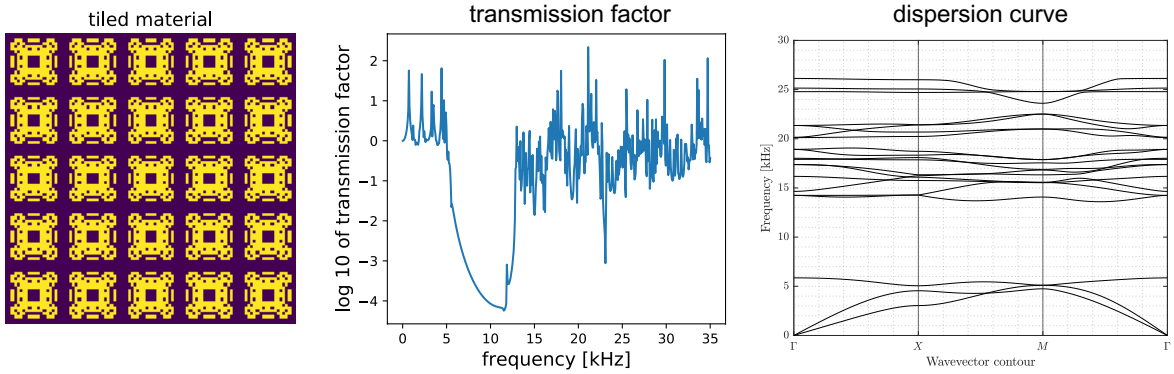
D.4 Finite Tiling COMSOL Test

We use a commercial finite-element solver [COMSOL, 44] to simulate the dispersion relations of finitely tiled materials. COMSOL is a standard, industrial finite element software tool. We set the label to 1 when the size of the band gap in the target frequency range was greater than 1 kHz; these large band gaps are useful in practice.

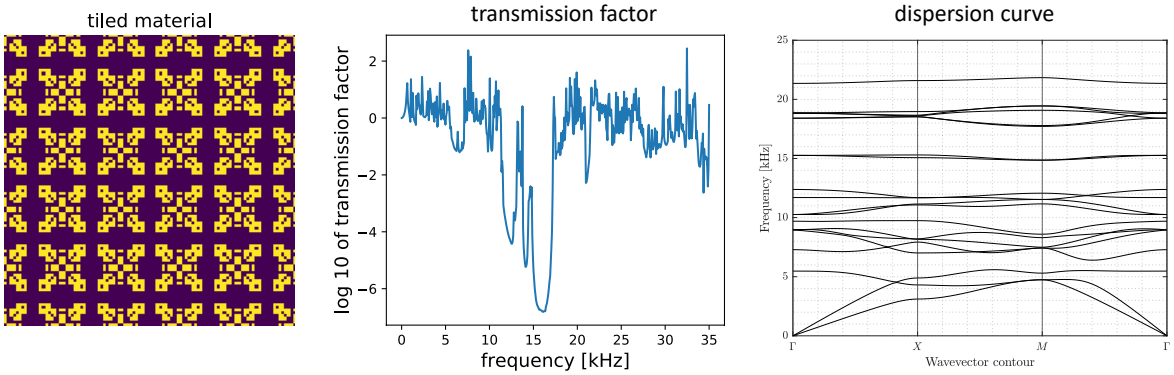
Figure S5 shows three examples of finitely tiled materials made by 20×20 unit-cells found by the unit-cell template sets, along with the wave transmission factors simulated by COMSOL, and their dispersion curves. The transmission factor is the ratio between the magnitude of the response on the right side of the material to the input signal on the left side of the material. Sharp drops in the transmission factor happen in the target frequency ranges in all three examples, indicating the existence of a band gap, which matches with the band gaps shown between the dispersion curves. In addition to the three examples in Figure S5, we tested the same finite tiling for 100 unit-cells in each frequency range, and all of them have the target band gap property. Therefore, our method is robust under finite tiling.

E Extension to Non-square Pixels and Unit-cells

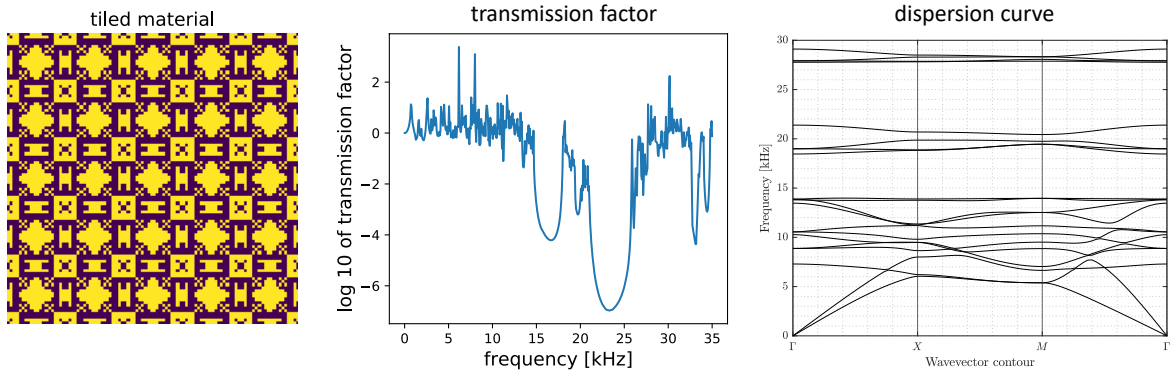
In the problem setting of our experiments, both the pixels and unit-cells are square. This is mainly due to the simplicity of simulation in square settings. However, the proposed methods are general enough to be adapted to nonsquare pixels and unit-cells. Our methods try to calculate or identify the existence of certain global or local patterns, but do not specify that their shape should be comprised of square pixels. As an example, Figure S6 shows how shape-frequency features and unit-cell templates can be adapted to nonsquare pixels (triangle pixel) and unit-cells (diamond unit-cell).



(a) Frequency range: [0,10] [kHz]



(b) Frequency range: [10,20] [kHz]



(c) Frequency range: [20,30] [kHz]

Figure S5: Finitely tiled materials and their corresponding transmission factor. *Left*: Tiled material (tiled 5×5 times) made by unit-cell designs selected by the proposed method with respect to three different frequency ranges. *Middle*: the transmission factor is the ratio between the magnitude of the response on the right side of the material to the input signal on the left side of the material. *Right*: the dispersion curves of the tiled material.

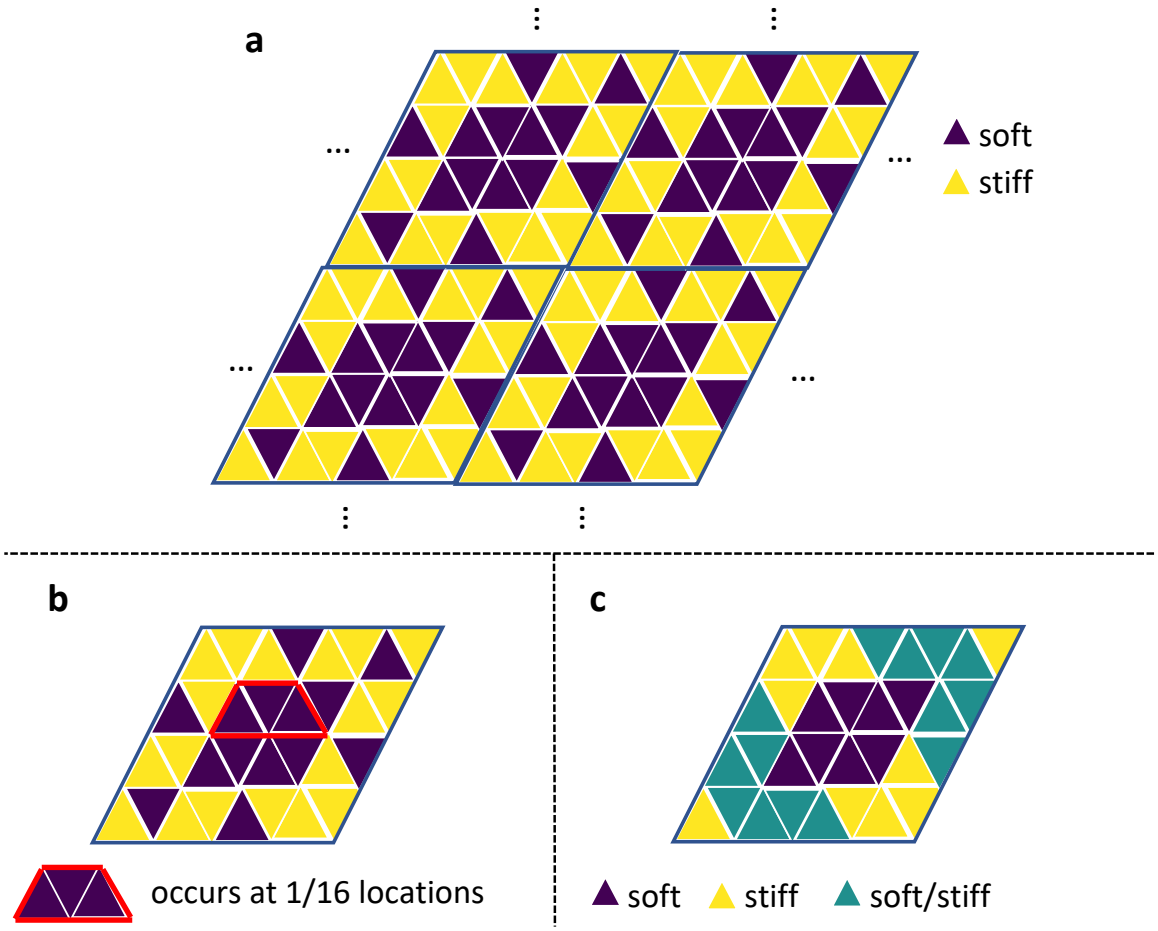


Figure S6: Nonsquare unit-cell and adaptation of proposed methods. **a** An example of metamaterial unit-cell design with triangle pixel and diamond unit-cell **b** An example of shape-frequency feature defined on this nonsquare design space; **c** A unit-cell template defined on this nonsquare design space that matches with the example unit-cell in **a**.